EE3EY4: Electrical Systems Integration Project
Lab 7: Localization and Mapping with MacAEV

Instructor: Dr. Sirouspour

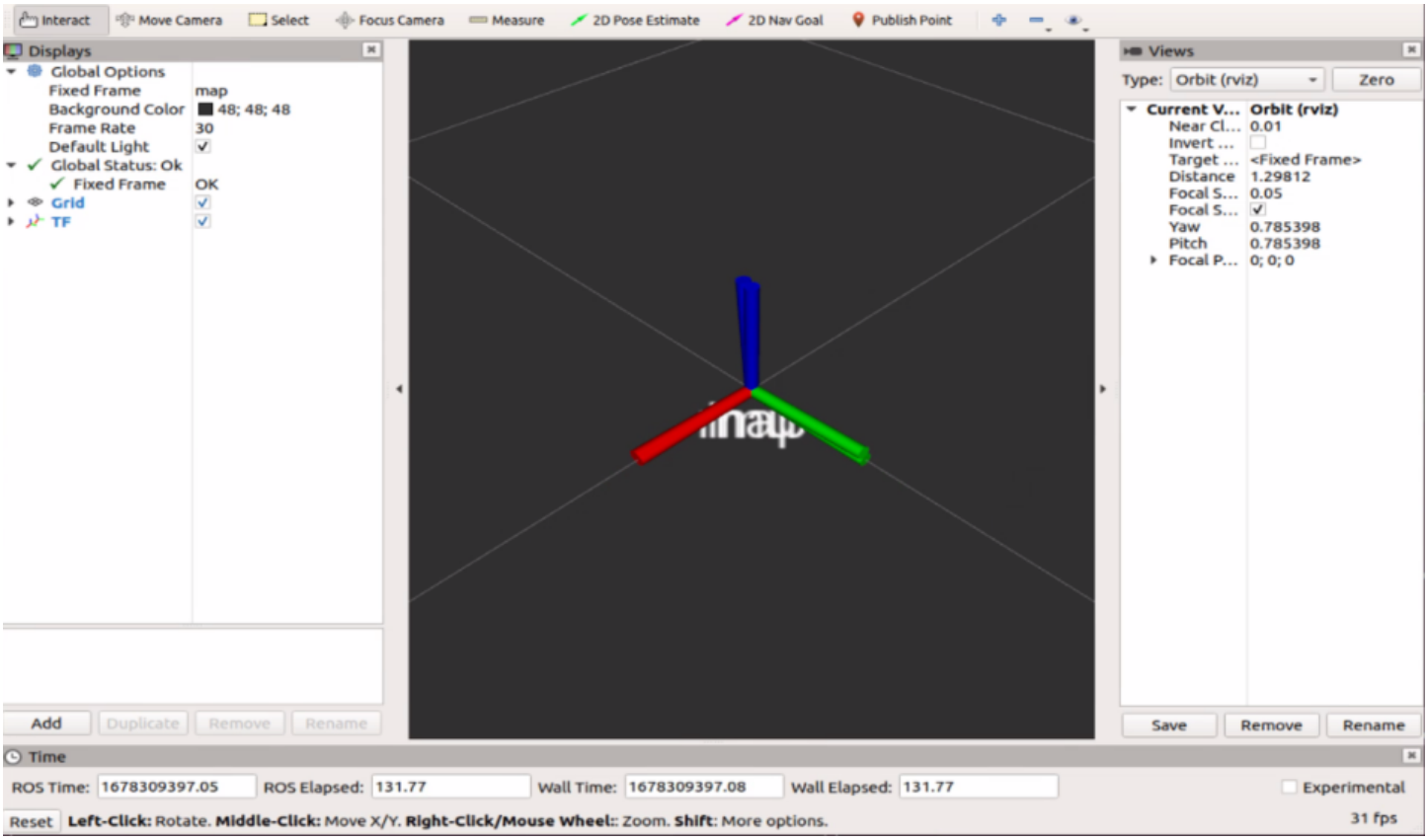
Aithihya Kompella - kompella - 400310794
Benji Richler - richlerb - 400296988
Shathurshika Chandrakumar - chands39 - 400315379
W. Tisuka Perera - L03 - pererw2 - 400318373

Date of Submission: March 22nd, 2023

Task: Complete the following ROS static transformation in the "experiment.launch" file that is provided to you on Avenue to Learn. The information given in Figs. 1&2 is needed for performing this task. Explain the purpose of these transformations.

The purpose of the transforms is to switch between different frames of reference using coordinate transforms. This is essential when keeping the data consistent between transformations. The first part of the code deals with transforming the baselink frame to the imu frame. The second line of code deals with transforming the baselink frame to the laser frame. This code is critical for the AEV to keep consistent data when performing localization and complete autonomous navigation with different frames.

Task: Use the following command to echo the measured orientation on screen and confirm that reported orientation is consistent with this convention (e.g., by aligning the y-axis of the IMU frame with the direction of the earth north):



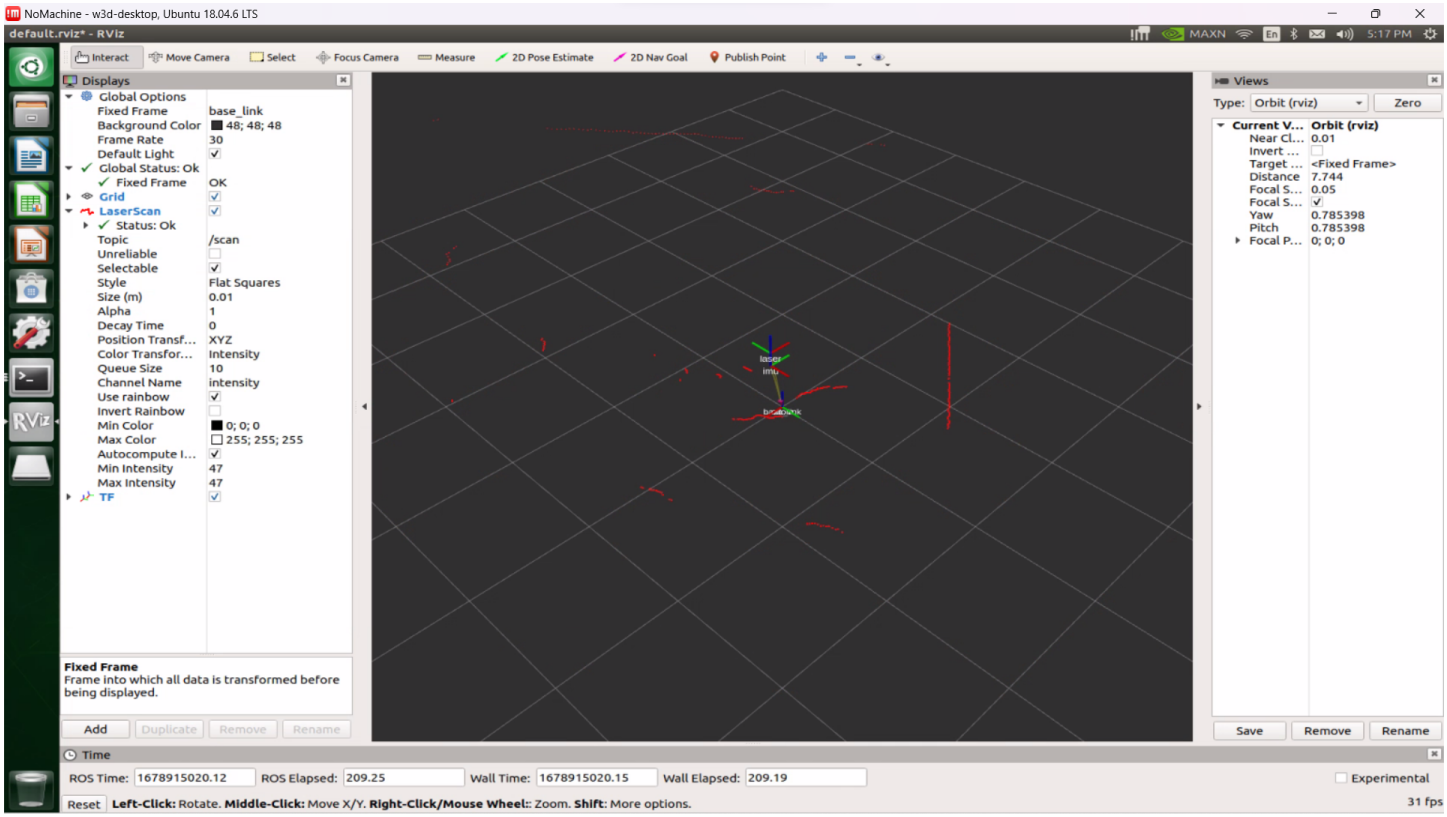
Task: Create a ROS package named imu_tf_pkg within your catkin workspace (see here for instructions) with the following dependencies: roscpp std_msgs sensor_msgs tf. Next, add the file imu_tf_pub.cpp as a node to your ROS package. Read the file's content and briefly explain in your report what the node does. Add the imu_tf_pub.launch file to the launch folder of the package, replace the CMakeLists.txt file in the package with the one you downloaded from avenue, and finally build the package

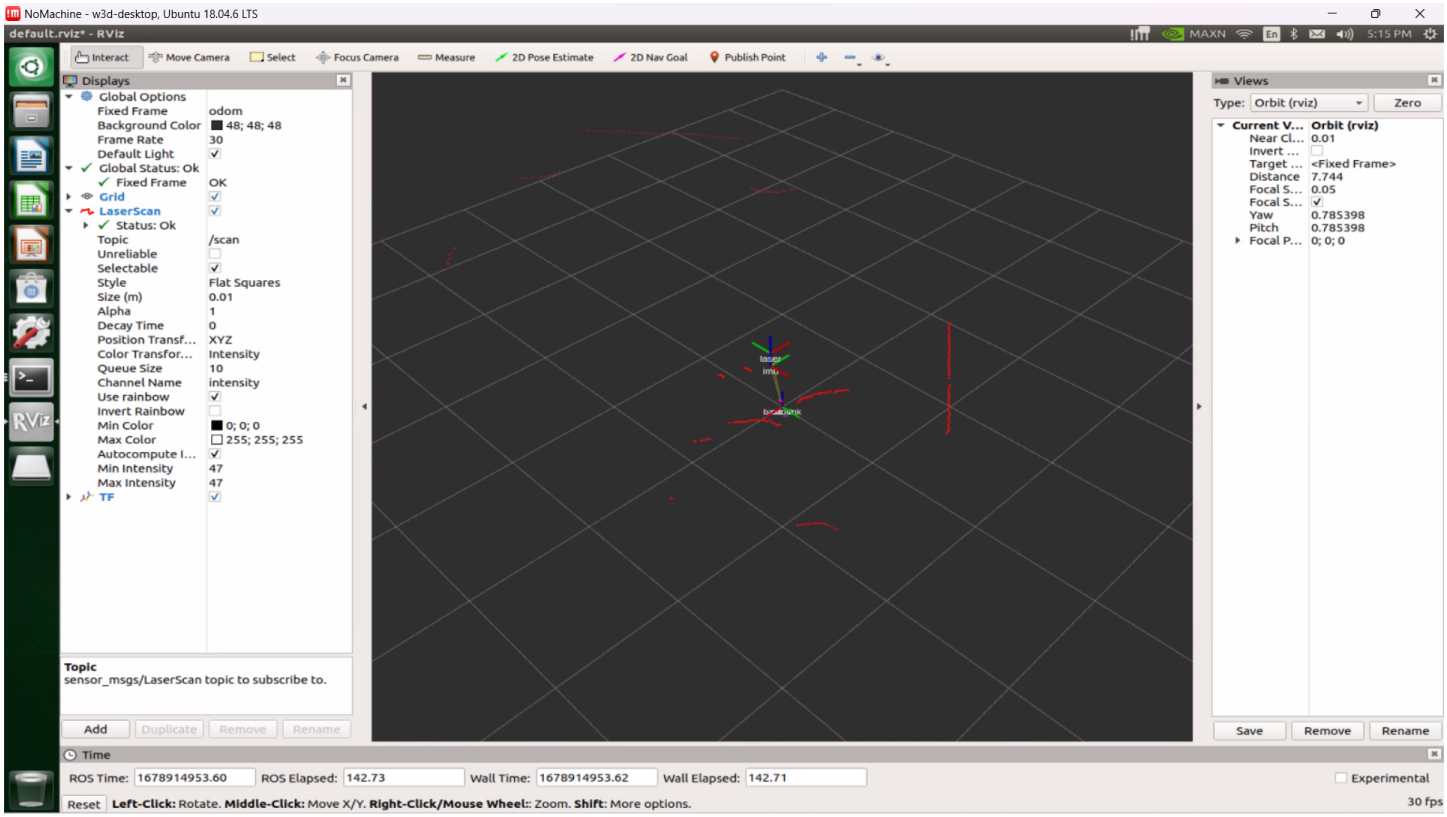
The imu_tf_pub node transmits the transformation between the parent frame (p_base_stabilized) and the child frame (p_base). As well as subscribing to the IMU topic, extracting the orientation quaternion from the IMU message, and converting it into a rotational matrix. This sets this as the rotation component of the two frame transformation.

Task: Discuss potential sources of error in wheel-based odometry computations that could lead to such drift in the computed position and orientation of the vehicle.

Potential sources of error could come from factors of estimation accuracy. This could potentially be frame misalignment, wheel calibration, and inaccuracies in the odometry model. External factors could be surface unevenness and wheel traction. Surface unevenness can lead to not allowing the AEV's wheels to not have direct contact with the ground, which could cause errors. Wheel traction or friction/slipperiness can also cause the possibility of over and underestimating the distance traveled by the AEV, which could lead to a drift in the orientation and position of the AEV.

Task: Add a "LaserScan" display type and set its topic to "scan". This should allow you to observe the laser scan data while driving the vehicle. Observe and compare the laser scan data in the two cases where the rviz fixed frame is set to "base_link" and when is set to "odom". Comment on your observations.





Setting the RVIZ fixed frame to “odom”, the laser scan data remains stationary as the vehicle moves. This is due to the fact that the odom reference frame is the fixed frame. When the RVIZ frame is set to base_link, the laser scan data moves as the AEV moves, meaning that the main reference frame in this case is the AEV, to which the base_link is attached at the middle of the rear axle.

Task: Complete the missing lines of code and uncomment the relevant sections in the file “experiment.cpp” to publish a new odometry message and a new ROS tf. Compare the results of vehicle odometry computation with and without the IMU angle measurement. This can be observed by comparing the frames “base_link” and “base_link_imu” in rviz visualization tool. You should set the rviz fixed frame to “odom”. Comment on potential advantages and disadvantages of each approach. Refer to Fig. 1 for the definition of the MacAEV frames. Explain how the code handles the difference in the frame of reference for measuring the yaw angle between IMU (EN-U frame) and the wheel odometry “odom”. Do you still expect to see error accumulation in the estimation of the vehicle states? Explain.

The onboard IMU directly measured the vehicle’s heading angle. In contrast, the approach without IMU uses the numerical integration of the angular velocity to estimate the position/orientation. When comparing the results between the two approaches, it was observed that without the IMU angle measurement, errors accumulated more drastically in the estimated vehicle state. This was evidenced by the base_link and base_link_imu frames, which showed that the former had a greater error in position and angle. With the base_link_imu frame it was noted that the estimation was almost exact, with the frame aligning back to the odom frame as expected when the AEV was driven back to the starting position.

The main advantage of the IMU angle measurement is the increased accuracy, as the orientation computation of the AEV is more precise. Over a longer period of time, IMU measurements will have little accumulated errors, creating a more accurate odometry computation. The disadvantage of the IMU approach lies in the increased expense and complex nature of the system when integrating the IMU sensor. In contrast, the approach without IMU is cheaper and has less complexity as there are no extra IMU sensors to add on to the system. The disadvantage to the lack of IMU is the error accumulation that comes with the estimation, meaning over time the estimated position/orientation of the AEV will drift.

The code calculates the yaw angle between the IMU (E-N-U frame) and the wheel odometry (odom) through the yaw angle from the IMU converted using a fixed offset value. A rotation matrix that relates the two frames is used for the conversion.

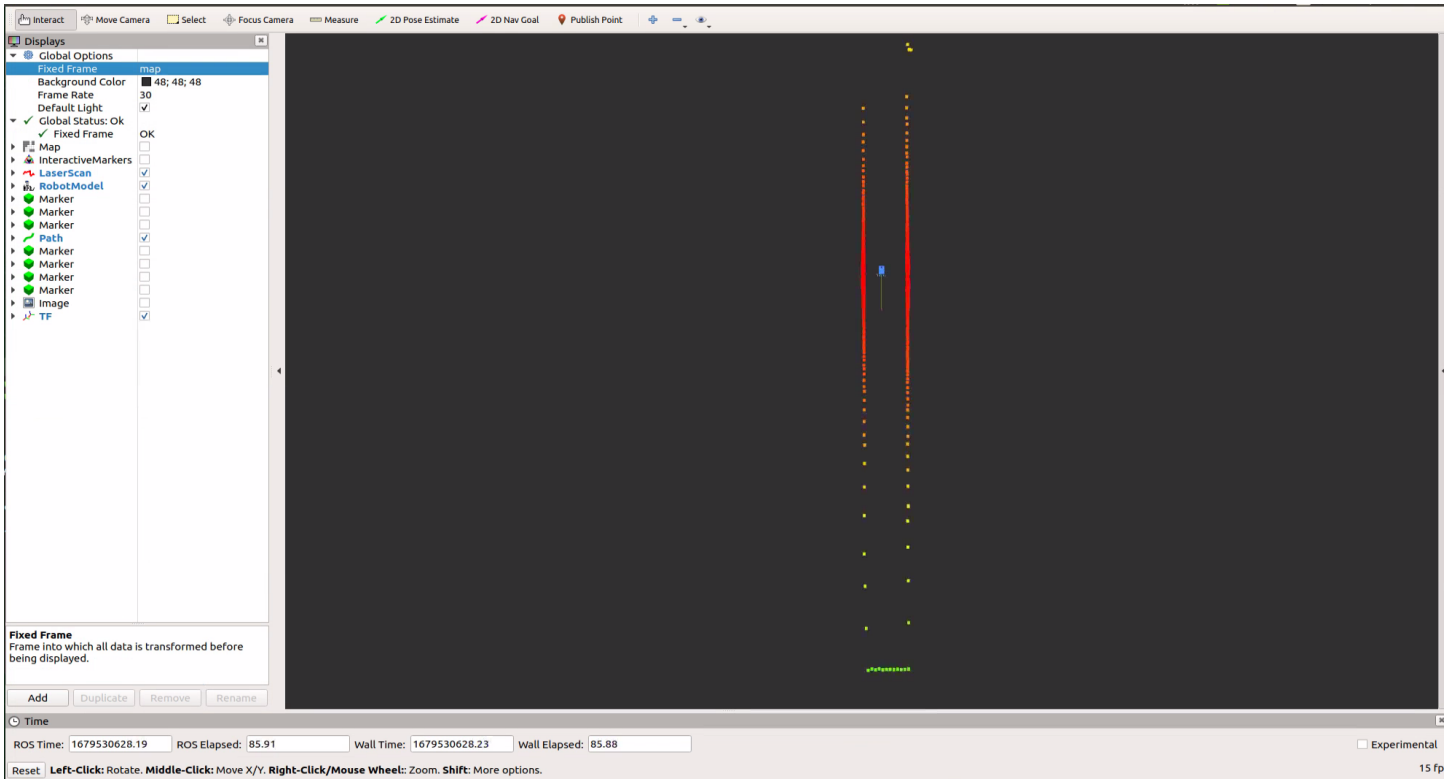
Error accumulation in the estimation of vehicle states is still expected, as the IMU data can still be affected by things like uneven terrain, friction and slippage of wheels. The IMU data serves to reduce these errors, and generally limits the error accumulation, which results in more accurate estimations.

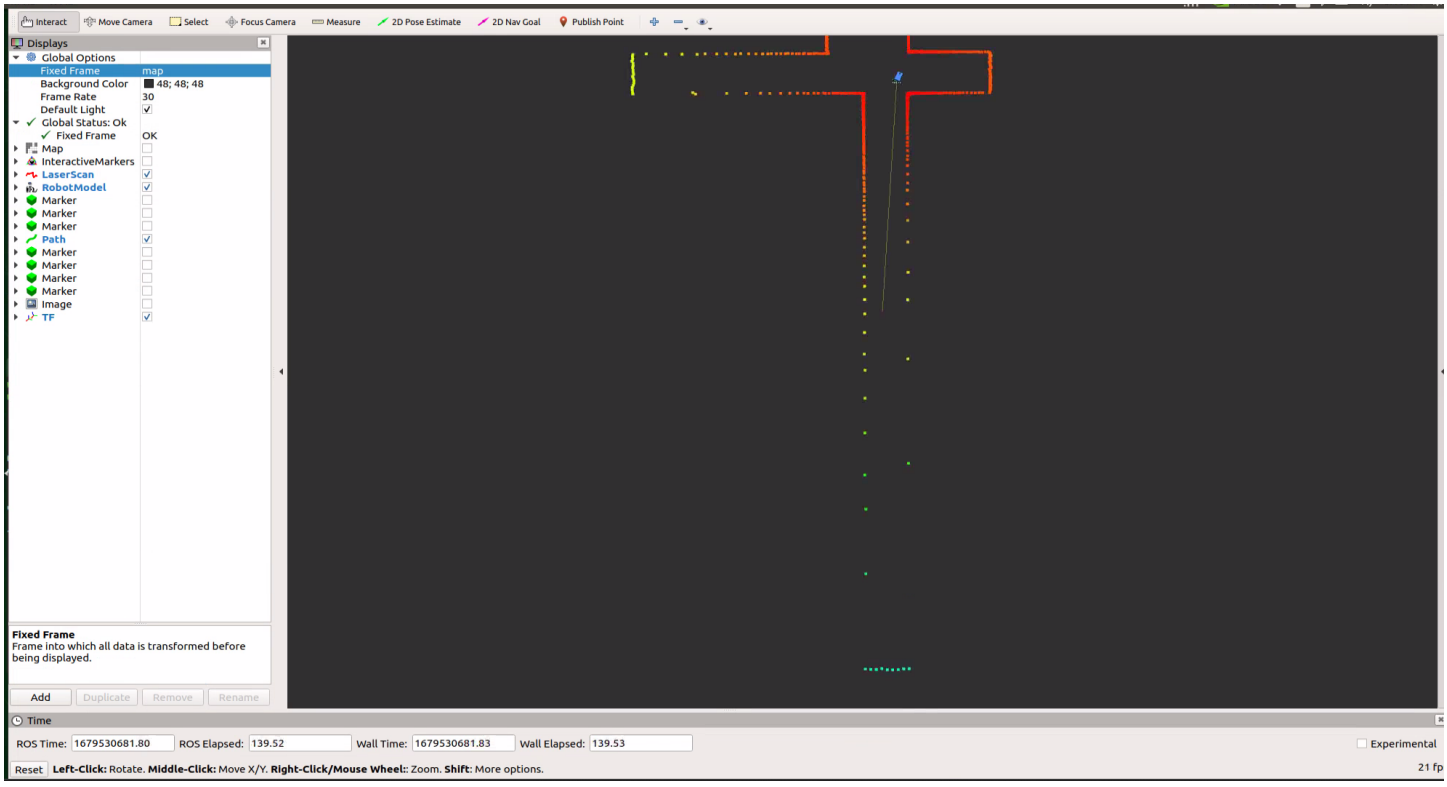
Task: In the launch file, the values of the two parameters "use_odom" and "use_imu" for the Scan Matcher are set to "true". Refer to the documentation and resources for the algorithm to explain what this means and how it could possibly help improve the accuracy of the localization.

When setting "use_odom" to "true" the Scan Matcher algorithm will use data from the wheel odometry to gauge the orientation and location of the AEV. When setting "use_imu" to "true" the Scan Matcher algorithm will use IMU data to gauge the orientation and location of the AEV. By using both parameters, it visualizes the surrounding area, the Scan Matcher algorithm can provide more accurate localization estimates of the AEV, as well as reducing the potential errors in the process. However, since the lidar sensor is implemented, there is a possibility of reading similar geometries as the same area. Another issue discovered with odometry, is that when the distance increases, the vehicle's localization functions start to malfunction.

Task: Launch the file "experiment.launch". Use the ROS visualization tool, rviz, to explore the performance of the localization algorithm. Set the fixed frame to "odom" and add a TF type display from the left panel. Also add a LaserScan type display with the topic "scan" to show the laser scan data. Observe how the frame "base_link" moves with the respect to the frame "odom" as you drive the vehicle around. Pay particular attention to the segments of the drive where the vehicle's surrounding environment may have uniform features, e.g., when driving in a long hallway where the main features are two uniform parallel walls. Do you expect any difficulty in SLAM when operating in such scenarios? Explain your answer. Comment on how odometry and/or IMU data may be helpful in such scenarios. Compare the performance of the Scan Matcher algorithm when using this data versus when not using them.

To test its functionality, the AEV was driven down a narrow hallway with two uniform parallel walls. Based on these constraints, it was challenging for SLAM to distinguish between the AEV's current location and its previous location. This is due to a lack of distinctive attributes not allowing for the localization and mapping functions to perform adequately and accurately. In this case, odometry along with the angle measurements of IMU will be beneficial when supplying data to the AEV in order to help measure the position, related to its previous position. The Scan Matcher algorithm's performance is also vastly improved when implementing odometry and IMU data.





Task: Refer to the documentation and reference paper for the Scan Matcher algorithm to understand the role of the parameters “kf_dist_linear” and “kf_dist_angular”. What are the potential trade-offs in selecting the values of these parameters?

kf_dist_linear (meters): the distance the fixed frame must travel before the keyframe scan is updated.
kf_dist_angular (radians): the angle the fixed frame must travel before the keyframe scan is updated.

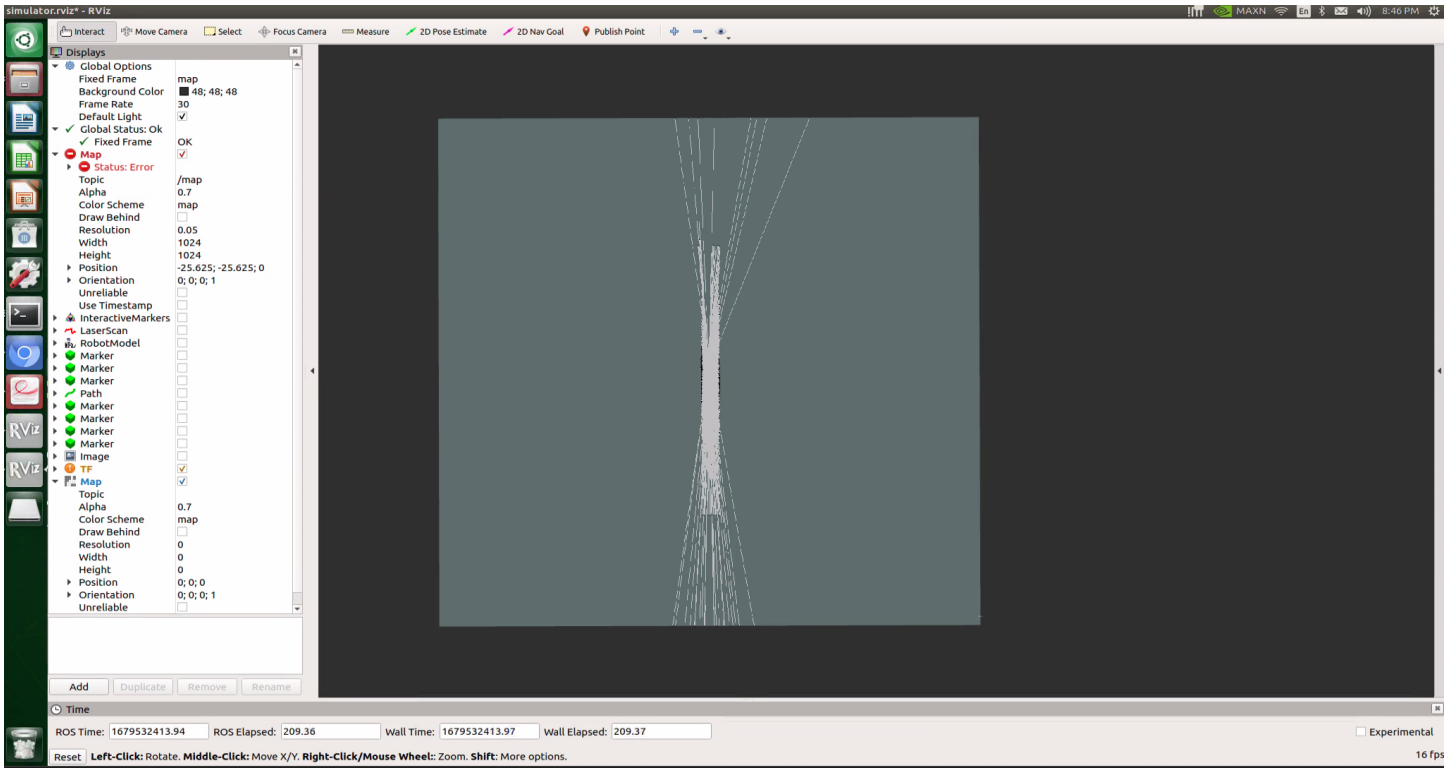
The tradeoff that comes with selecting these parameters is that if the values are set very low, the localization estimates are more accurate, but the frame-to-frame scan matching rate is very high. Conversely selecting high values for these parameters results in greater inaccuracies in the estimations, but with decreased scan matching rate. Essentially the tradeoff is between accuracy and computational efficiency.

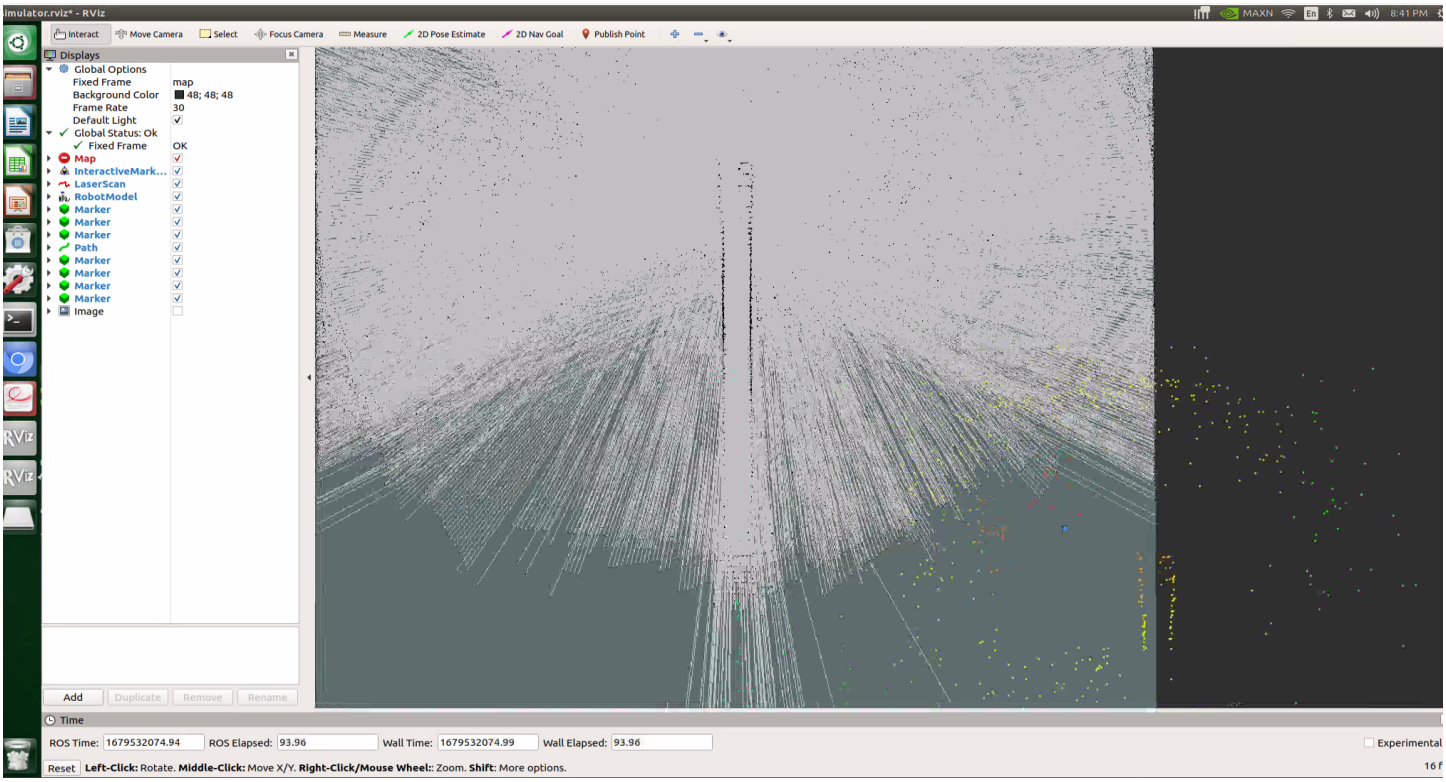
Task: Compare the localization results of the Scan Matcher algorithm to those obtained by the combination of wheel odometry and IMU data in the previous section.

The location of the AEV was determined using the IMU and wheel odometry data, collected from the controller inputs and internal gyroscope. The IMU and wheel odometry moves back and forth and rotates in tangent with the AEV. The scan matcher uses the surrounding environment to determine its location, the laser rotating when the car rotates.

Task: Ensure that the vehicle surrounding is clear of objects and people as you will be driving the vehicle using the Joystick. Modify the launch file and run it from the command line to launch the nodes and start the SLAM algorithm. Also launch the ROS visualization tool rviz to display the results of the SLAM algorithm. Set the rviz fixed frame to "map" and add displays of TF and Map types from the left panel. Deselect all frames except for "map" (fixed frame) and "base_link" (body attached frame) under TF. To display the map data, add "map" under the topic option of Map display. Drive the vehicle around slowly and observe how the map of the environment is constructed while the vehicle is simultaneously localized in this map. Comment on what you observe.

Task: The hector SLAM lacks a loop closure mechanism. This simply means that if you revisit a part of the environment that you have seen before, the algorithm would not be able to recognize this and it could produce erroneous results. You should try to observe this potential issue by driving the vehicle around and then returning to areas that you have explored before. Save a copy of the construct map for your report. You can use the "map_saver" for this purpose. See here for information about how to use the command.





When the AEV has moved minimally, the map generated is seen in the first image. This map shows the environment of the AEV as two straight walls indicated by the black outline. As the walls extend beyond the sensor's reach, the map generated becomes more distorted, as seen in the radial lines that extend outwards near the top and bottom of the first image. The second map generated is the environment constructed when the AEV has moved a greater distance forward and back. This map is clearly distorted as the SLAM algorithm does not have memory, meaning it cannot recognize objects it has already scanned. This is clearly seen in the second image where the map is heavily distorted, with copies of the hallway overlaid on top of one another.

Task: Examine the effect of the two parameters "map_update_distance_thresh" and "map_update_angle_thresh" on the performance of the hector SLAM algorithm. Explain what these parameters do.

The parameter map_update_distance_thresh represents the threshold for distance that the vehicle travels before the map updates itself. Similarly the parameter map_update_angle_thresh is the threshold for angular change before the map updates. The smaller these parameters, the greater the accuracy of the SLAM algorithm, as the map updates more frequently and for smaller magnitudes of change.

Task: Provide a summary statement reflecting on what you have learned throughout the activities of the past two weeks in localization based on wheel odometry and using LiDAR information for localization and mapping.

First, the group learnt how to convert between different frames of reference such as imu, laser and base_link through transformation matrices. The importance of converting data between these coordinate frames during odometry computations was realized in this step. Next, vehicle odometry computations were completed through two different methods of numerical integration of angular velocity, and adding IMU angle measurements. It was found that including IMU angle measurements greatly increased the accuracy of the vehicle state estimations, although it came at a drawback of increased cost and complexity of the system. Next, the 2D maps produced from vehicle odometry were created using the LiDAR sensor. This data was fed into the Scan Matcher Algorithm Laser Scan Data, which localized the AEV's surrounding environment by computing the position of the vehicle relative to change in position in its surroundings. It was found that this method had limitations in uniform surroundings like hallways, where the AEV did not compute a change in

position as the surrounding environment was believed to be static. Another method of Hector Simultaneous Localization and Mapping (SLAM) was explored. This algorithm continuously updated the map of the environment while also computing the position of the vehicle relative to the environment. The limitation found was that the algorithm held no memory of the environment, thus scanned objects will not be remembered. This has its own errors that result in inaccuracies in the 2D map created, as shown in the images above.

-END-