Development of a localization solution

The problem statement consisted of outdoor localization of a non-holonomic drive (car) by fusing data from speedometer, IMU and GNSS (GPS).

Approach –

Utilize non-linear state estimator – Extended Kalman Filter (EKF) which provides a method of estimating the state (pose of the car) by incorporating data from Odometry, IMU and GPS. For this purpose, off the shelf package of **robot_localization** in ROS, was used to fuse sensors and localize the car in the frame of the GPS. Utilization of the package comes with a cost of formatting the data and correcting the frames, as it is a generalized package for both 2D and 3D cases.

Methodology –

1) Creating publisher node for different sensor data type- robot_localization package in ROS requires us to provide the data in a particular format (ROS message).
    a. For the speedometer, the data was formatted as nav_msgs/Odometry.
    b. For the IMU, the data was formatted as sensor_msgs/Imu
    c. For the GNSS, nav_msgs/Odometry was used, contrary to commonly used sensor_msgs/NavSatFix, as the data was reported in the local inertial frame instead of global latitude and longitudes.
    d. Since, the sensors had different frequencies care was taken to ensure that the sensors reproduce the data as in real time. The time reported by IMU which had the highest frequency was used to ensure synchronous data publishing.
    e. Covariance value which plays an important role in the EKF, was selected approximately.

2) Pre-processing the data to appropriate frames – robot_localization utilizes 3 frames – map, odom, base_link to handle the data subscription and also report the estimated pose. Map frame is the frame which reports the filtered estimate. Odom_frame is the initial position of the robot and the base_link frame moves with the robot which is a result of transformation of the estimated pose w.r.t the odom_frame. In this case, odom_frame and map_frame coincided and a static tf publisher was required to publish this information. Base_link to odom transform was published by the robot_localization package, after an initial unsuccessful attempt to create a tf_broadcaster.

3) Configuring robot_localization package – Since the package is generalized, it provides a user with multiple parameters to configure the filter. Various parameters were studied from this link. Also, since system model was that of a car, 2D estimation mode was selected to reduce the error in estimation due to inaccurate IMU measurements. Parameters configured are listed as – frequence, two_d_mode, publish_tf (true), frame_references, topic_refrences to appropriate name as given by the publisher node etc.

4) Integration of GPS (discrete data) along with IMU and Speedometer(continuous data) – The package allows us to fuse discrete data from GPS which made jumps to that of continuous differential data stream coming from IMU and Speedometer. For these, two instances of robot_localization was launched – 1) First one fused Speedometer and IMU data to give a local

estimate in the odom_frame. 2) Second one utilised the filtered continuous data along with GPS data to give final state estimate

5) Correction of data frame. This was one of the major challenge in the task as the frame in which the measurement was taken was not known. After iterative correction, it was found the measurements were taken in NED frame (North, East and Down for X, Y and Z) whereas the robot_localization requires its data to be published in ENU frame (East, North and Up for X, Y, and Z). Fig 1 shows the result of publishing data in the incorrect frame. IMU, Speedometer and GPS data were corrected to get reasonable estimate. Further to coincide the GPS data with the estimated filtered state, initial heading was calculated approximately from the initial GPS data.

   Fig 2. presents the rqt_graph of the system in ROS and Fig 3. presents localization result

Further questions –

1) What can happen if 1 measurement is delayed ?
   The system will continue with the approximated future state and take it as measurement. This is a principle advantage of using Kalman Filter. The results show that GPS data is not continuous and makes discreet jumps, however the system is able to generate estimated pose in absence of it. However large delay may cause the car to drift considerably as the Kalman will not get any measurement to correct its gain.

2) Now your IMU gets damaged. How your implementation deals with it?
   IMU is a crucial component of estimation as it only can provide an estimate of the heading. In the case where the IMU is damaged, the car will try to correct its position once the GPS data comes in, however the correct orientation is not guaranteed. The package provides utilization of multiple redundant IMU's as failsafe.

3) How will you deal with the case when the car is in a place where it has been before, but there is an error and localization is showing a different place ?
   This is a challenging problem in the field of localization and SLAM – loop closure. Based on my limited research, visual odometry (camera, laser) is required to generate BoW (bag of words feature) to estimate whether the car is in a location before. Unfortunately, conventional GPS sensors are not accurate enough to provide this information. Few solutions have been presented using geometric and graph based approach, however the practical robustness has not been tested. The present solution using ekf_localization package has no means of estimating this information.

Further improvements –

1) Tune the covariance measurement matrices for odometry data, GPS and IMU.
2) Understanding the significance of HDOP and VDOP in the GPS data
3) Implementation of particle filter, since it can accommodate non-linearity in the system.
4) Due to high frequency rate of IMU, robot_localization often published random pose in between (maybe due to low computational power), this can be prevented by reducing the publishing rate and averaging IMU data.

References –

1) http://docs.ros.org/melodic/api/robot_localization/html/state_estimation_nodes.html (last accessed - 17th September 2018
2) Moore T., Stouch D. (2016) A Generalized Extended Kalman Filter Implementation for the Robot Operating System. In: Menegatti E., Michael N., Berns K., Yamaguchi H. (eds) Intelligent Autonomous Systems 13. Advances in Intelligent Systems and Computing, vol 302. Springer, Cham
1) Hasan, Md Maruf Ibne, Indoor and outdoor localization of a mobile robot fusing sensor data, Thesis, Boston, Massachusetts : Northeastern University, December 2017

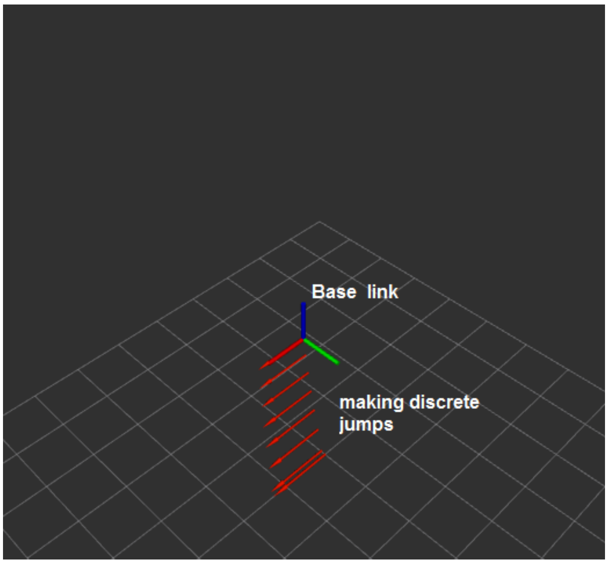

Fig. 1 Error due to ambiguity in frame.

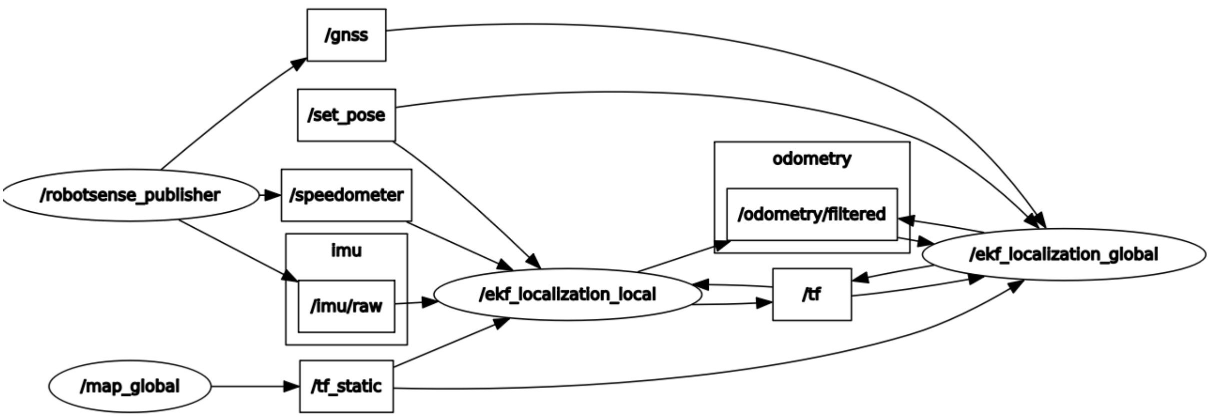

Fig. 2 Rqt_graph of the node

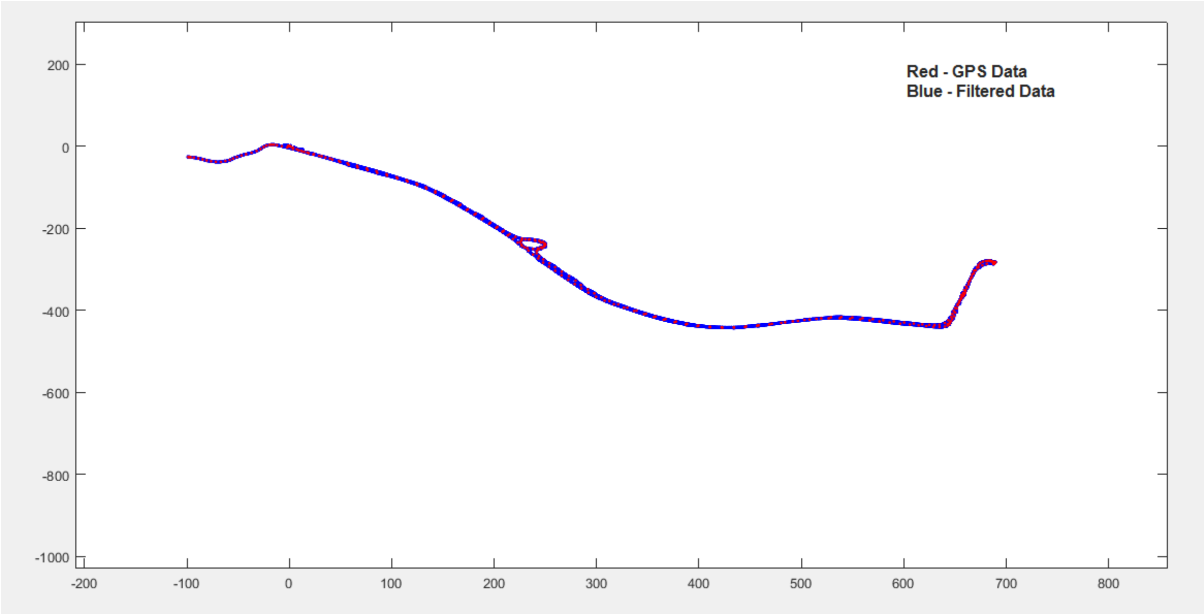

Fig 3. – Localization Result (Erroneous point due to high frequency were removed)