**Unscented Kalman Filter**

We are implementing the Unscented Kalman Filter on the data we are getting from IMU and processed data from the camera on the quadrotor. The data from the camera on the quadrotor is used as measurement data in the Unscented Kalman Filter.

We are using IMU data - angular velocities and linear acceleration for the prediction step in this filter.

There are two steps in this filter Prediction and Update and each part of the project it is implemented in a different way.

In both parts of the project, the prediction step has non-Additive noise

**Part 1 –** In Part 1 of the project, we are using the pose of the robot (position and angle of the robot with respect to the world frame) as a measurement in the update step.

First, we have to define xaug = with mean µaug = and covariance ∑aug =

**Step 1: Compute the Sigma Points**

We have to choose a set of 2n’ + 1 sigma points using the ith column of the Cholesky Decomposition of ∑aug .



Here α = 0.001, n’ = n+nq, n = 15 and nq = 12, k = 1

**Step 2: Propagate the Sigma points**

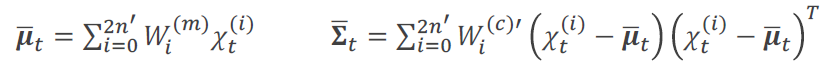
We have to create a process model a nonlinear function which will be used to propagate the sigma points to predict the new points.

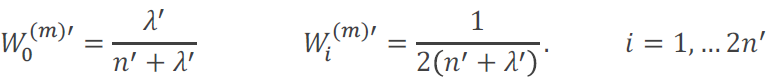


A picture containing graphical user interface

Description automatically generatedHere Function “f”

We have to add the results from function to the respective sigma points to calculate the predicted mean.

**Step 3: Computing predicted mean and covariance matrix.**



A picture containing diagram

Description automatically generated

So, by using above methods we can calculate the predicated mean and covariance which can be used for Update Step.

**Update Step**

****

We have to use the above equations to perform the Update step. Here zt measurements are coming from the pose data coming from the camera mounted on the quadrotor from Project 2.

**Part 2 –** In Part 2, we are using optical flow velocity (linear velocity in the camera frame) data as zt in the measurement update. We are also assuming the angular velocity coming from the zt measurement and the predicted angular velocity is the same. So, the update step is using the linear velocity of the camera in the world in the camera frame as a measurement update.

In Part 2 the Prediction step is as similar to the prediction step in part 1. i.e. Steps 1 to 3 in part 1 is as the same as in Part 2.

There are the following steps in the Update step. We are doing this step according to the additive noise.

**Update Step**

Here Rot = [cosd(45) -sind(45) 0 % Rotation matrix representing

-sind(45) -cosd(45) 0 the body in Camera frame.

0 0 -1];

Trans = [-0.04

0 % Translation vector from Camera to Body Frame

-0.03];

Here g(xt) =

x2 =

Text, letter

Description automatically generated

l(x2,x3) = Rot × transpose( ) × linear velocity(x3)

****

Here zuk is zµ,t

After doing the above three steps, I am computing the filter gain, filter state mean and covariance by using the formulas mentioned below.

A picture containing text, clock, orange

Description automatically generated

So, this is how, I am doing the Part 1 and Part 2 of the project.

**Results, Plots & Conclusions –** Now, I am attaching the graphs coming from the results of both Part 1 and Part 2.

Chart, diagram

Description automatically generated

Part 1 Dataset 1

Chart

Description automatically generated

Part 1 Dataset 4

When we look at the plots of both datasets 1 and 4 for part 1 of the project, we will find that the blue and red plots are approximately matching.

For dataset 1, the accuracy of plot matching is quite better however for dataset 4, some plots for position, velocity is not so good. One of the reasons are the parameters like na, ng, nbg, nba and R are not tuned in a better way and there is some error in dataset 4.

Chart, diagram

Description automatically generated

Part 2, Dataset 1

Chart, diagram

Description automatically generated

Part 2, Dataset 4

When we look at the plots for both Dataset 1 and Dataset 4 of Part 2, we will see that the plots in blue and red are not quite overlapping as compared to Part 1 of the Project. The major reason for this is the tuning of some values of na, ng, nba, nbg, and R. If these variables can be tuned then the plots may overlap with each other, if I had more time, I would have done that, but I tried my best to make both blue and red plots converge by tuning the values na, ng, nba, nbg and R. It was a great learning experience.