ROBOT LOCALIZATION AND NAVIGATION (ROB-GY 6213)

PROJECT 2 - REPORT

NAME: ***Andrew Devadhason*** *Net ID:* ***ad7068***

PROJECT -2 TOPIC: OPTICAL FLOW BASED POSE ESTIMATION AND TRACKING

TASK:

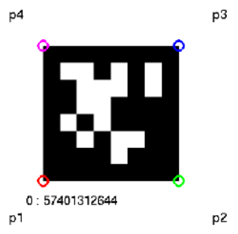
In this project, based on the given handout, the goal is to implement a vision-based 3D pose estimator which estimates the position, orientation, linear and angular velocities of a Nano+ quadcopter. The drone is made to fly over a mat of April tags and the data relating to the image is provided. The project initially involves implementation of projective transformation-based position and orientation estimation and then involves calculation of velocities through optical flow based on the given image data where RANSAC would be implemented for outlier rejection.

PROCESS:

The project is split into three parts involving sequential implementation of the process involved in the task to be performed. In the following, the approach involved in each part will be explained.

PART-1:

The process in part-1 of the project-2 involves estimation of position and orientation of the drone from 2D points in the image and 3D points in the world of the April tags. As the initial input parameters, the camera intrinsic matrix, the rotation and translation information from the camera frame to body frame and the sequential matrix of all the ids of April tags in the world.



- The initial step is to extract all the points of the April tags based on the tags detected in the current image in the current timestamp.
- The next step is to compute the coordinate points of the April tags with respect to the world frame based on the information given in the handout which would be used in the function getCorner().
  ```
  function res = getCorner(id)- res has the center,p1,p2,p3 and p4
  of the tag
  ```
- Then the A matrix must be constructed to form the overdetermined equation of matrix nX9 where n is the total number of equations. Each point requires two equations. The unknowns in this case is the h matrix which is obtained by solving the A matrix through SVD.

$$\begin{pmatrix} x_i & y_i & 1 & 0 & 0 & 0 & -x_i'x_i & -x_i'y_i & -x_i' \\ 0 & 0 & 0 & x_i & y_i & 1 & -y_i'x_i & -y_i'y_i & -y_i' \end{pmatrix} \underset{\uparrow}{\boxed{h}} = 0$$

- The 9th column of the 9x9 V matrix obtained through solving the SVD is the transformation of the projection.
- Furthermore, the H matrix is multiplied by the inverse of the camera intrinsic matrix to make it unbiased.
- Then optimization must be performed to satisfy the following constrains. This is performed by performing SVD again over the obtained rotation matrix. The translation is obtained by normalizing the last column of the H matrix with the first column of the Rotation matrix.

$$R_1^T R_2 = 0$$

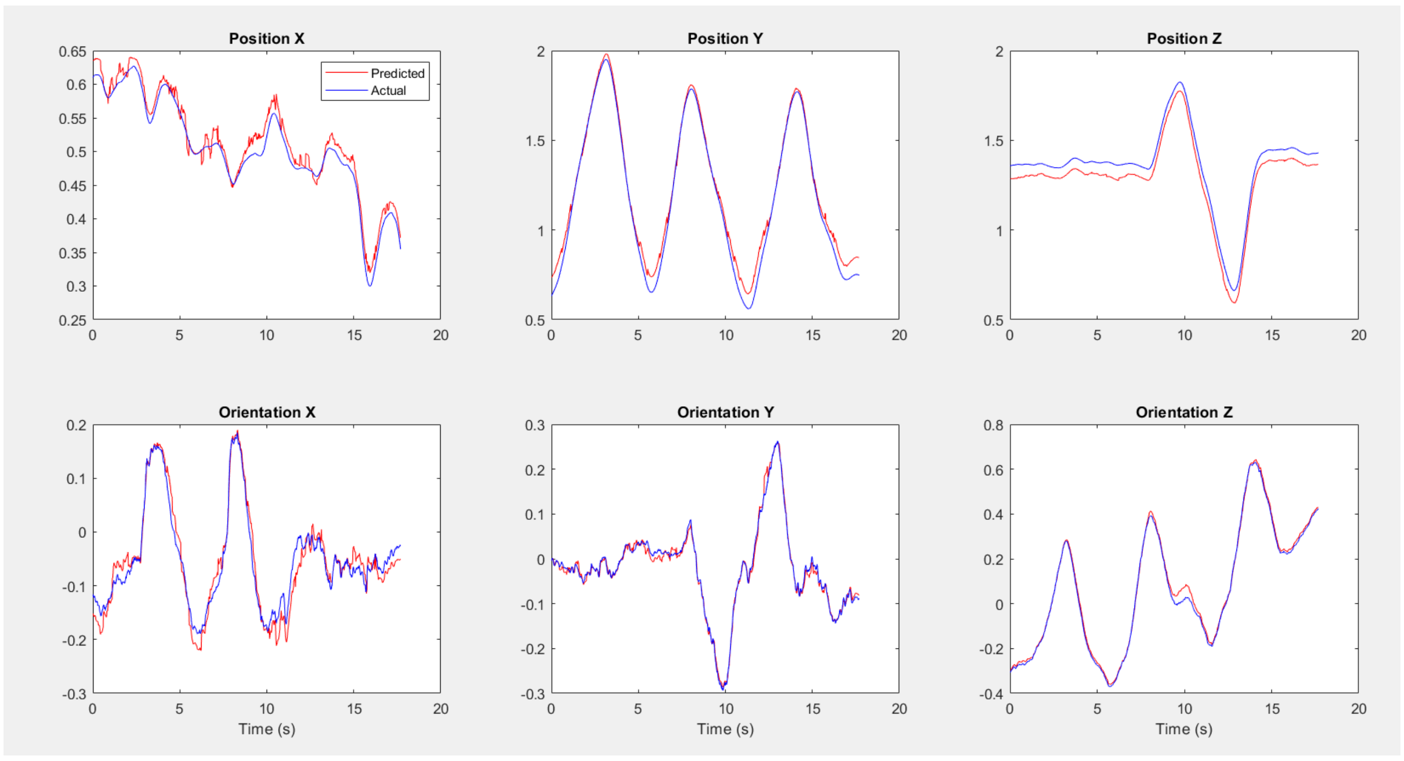
$$\|R_1\| = \|R_2\| = 1$$

$$T = \hat{T} / \|\hat{R}_1\|$$

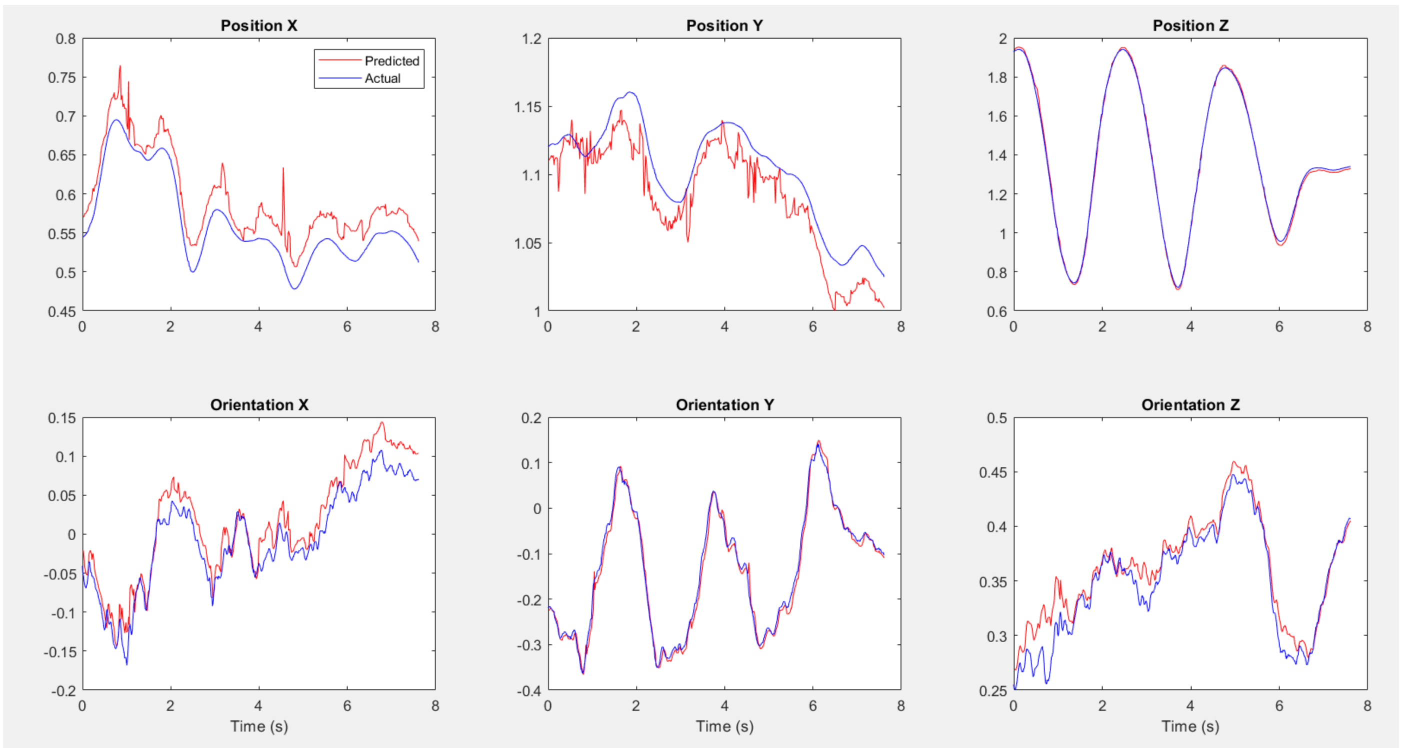
- Finally, a transformation is performed to transform everything to the body frame of the drone based on the given information in the handout.

RESULTS- PART-1:

- The results from part one of the project are conclusive with the estimated position and orientation aligning well with the actual data with minor bias and noise factors due to jitter in time flow.



P1: Outcomes of Pose Estimation for Dataset-1



P1: Outcomes of Pose Estimation for Dataset-4

PART-2 & 3:

The process in part-2&3 of the project-2 involves estimation of the linear and angular velocity of the drone through the optical flow of the image data. The results from part 1 which include the estimatePose function and the getCorners() function are reused for Part-2 of the project since the position are taken as inputs.

- The first step involves initialising all the required variables which include the camera intrinsic matrix, the transformation matrix etc. Further the timestamps are passed through a low pass filter to remove any jitter for further calculation.
- The next step is to calculate the optical flow between the current and previous image. For this the Harris corner detector is used from the Matlab function to detect the corners of the April tags in the image data.
- Through ***vision.pointTracker()*** the next point is tracked and then normalized with the camera intrinsic matrix.
- If the RANSAC Flag is enabled then the p_dot is calculated and the velocityRANSAC() function is called with all the input parameters mentioned in detail with comments in the code.

$$\dot{\mathbf{p}} = \frac{1}{Z}A(\mathbf{p})\mathbf{V} + B(\mathbf{p})\mathbf{\Omega}$$

$$\left(\begin{pmatrix} \frac{1}{Z_1}A(p_1) & B(p_1) \\ & \vdots \\ \frac{1}{Z_n}A(p_n) & B(p_n) \end{pmatrix}\begin{pmatrix}\mathbf{V}\\ \mathbf{\Omega}\end{pmatrix} - \begin{pmatrix}\dot{p}_1\\ \vdots \\ \dot{p}_n\end{pmatrix}\right)^T \left(\begin{pmatrix} \frac{1}{Z_1}A(p_1) & B(p_1) \\ & \vdots \\ \frac{1}{Z_n}A(p_n) & B(p_n) \end{pmatrix}\begin{pmatrix}\mathbf{V}\\ \mathbf{\Omega}\end{pmatrix} - \begin{pmatrix}\dot{p}_1\\ \vdots \\ \dot{p}_n\end{pmatrix}\right)$$

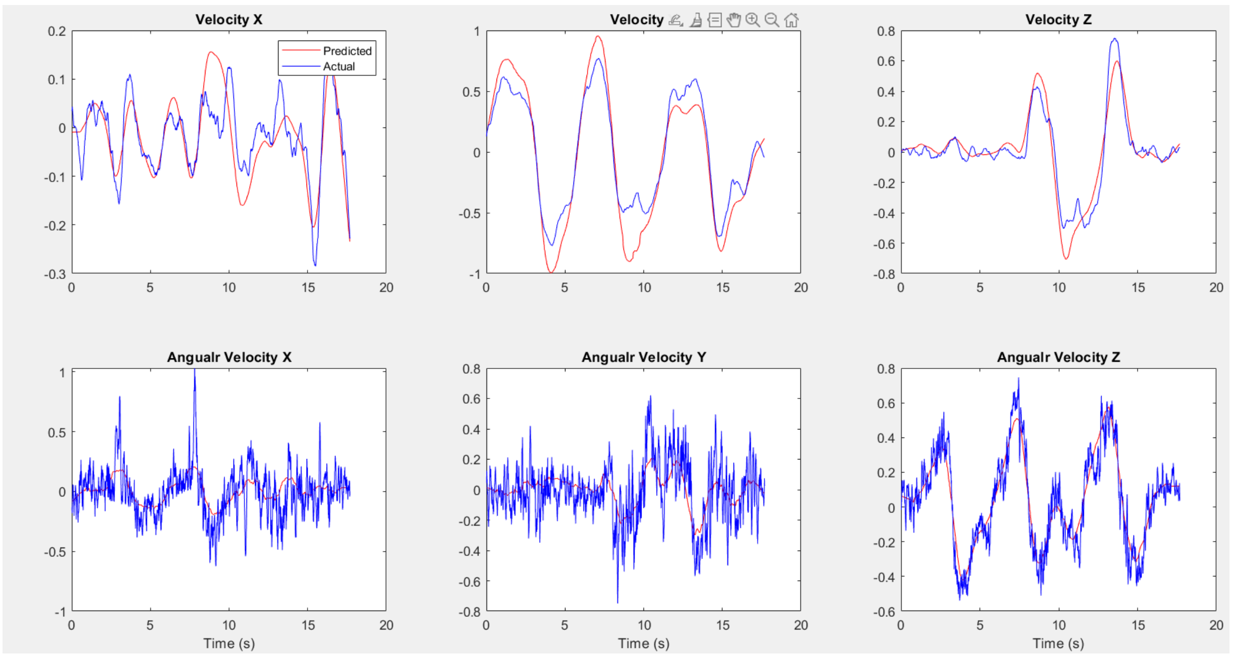
- In the RANSAC function, the H matrix is formed with M number of random points in our case M is chosen to be 3. Then the velocity is computed for the chosen points and further, the inliers are counted in this set based on which the Velocity of certain points are taken into consideration for the final estimate and the outliers are rejected.
- In the RANSAC function the value of the number of iterations k is calculated based on the following according to the lecture:

$$k = \frac{log(1 - p_{success})}{log(1 - \epsilon^M)}$$

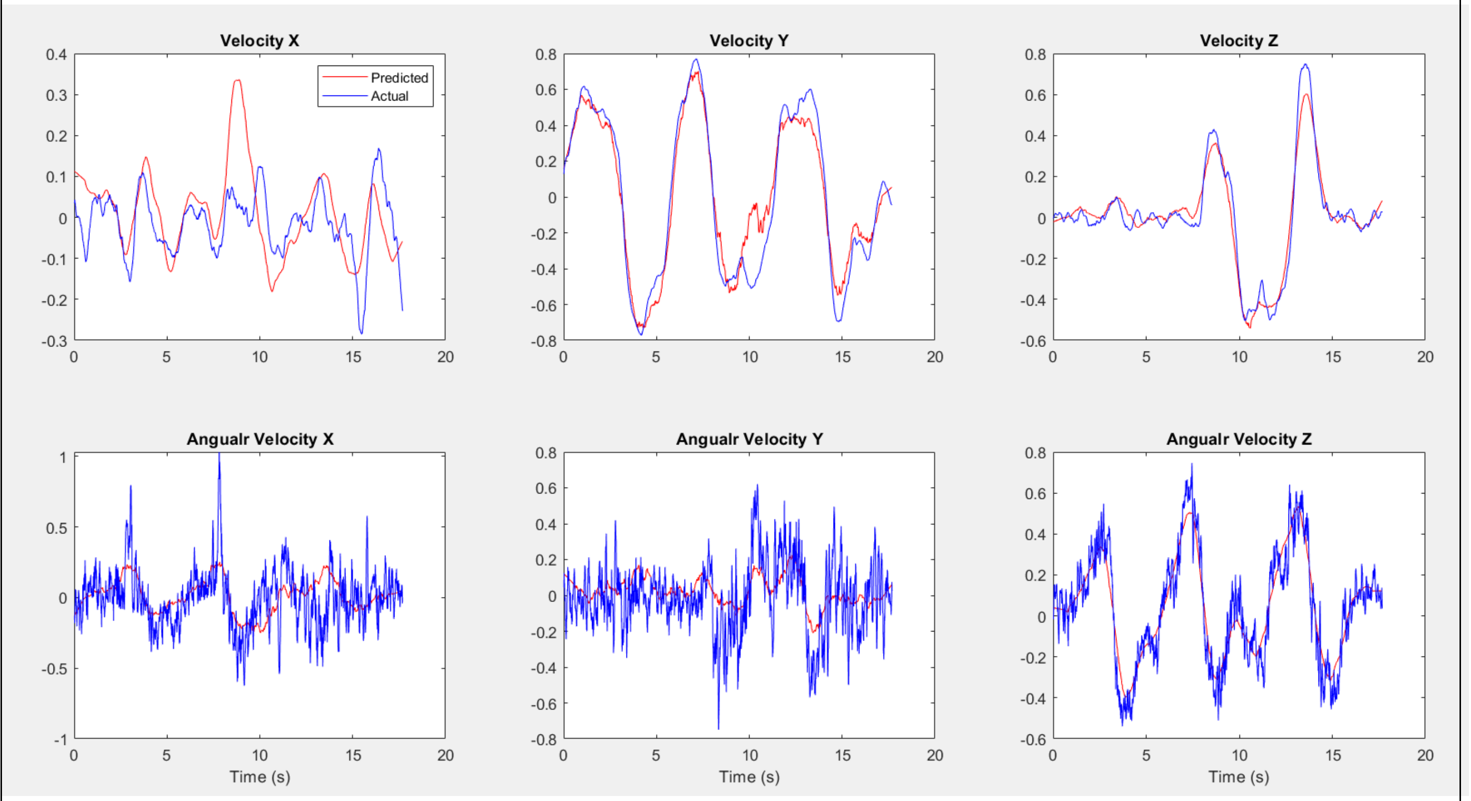
- If the RANSAC Flag is disabled, the Ap and Bp matrices are formed and further the H matrix is formed for which the pseudoinverse is taken and is further multiplied with the p_dot.
- This is transformed from the camera to the body frame and the final V and W are obtained.
- These final values are passed through a low pass filter to better improve the quality of estimation.

RESULTS- PART-2&3:

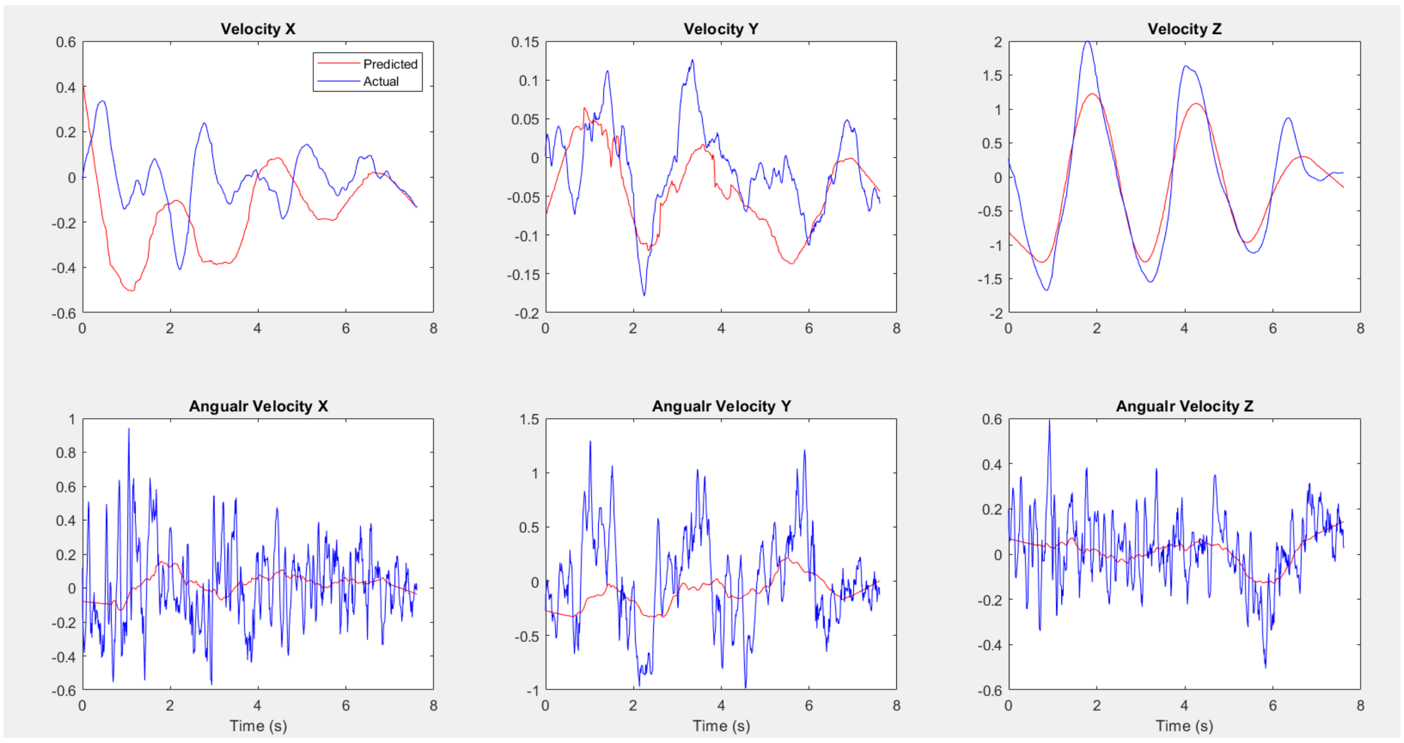
- The results from part two are split into two segments. With and without the RANSAC based outlier rejection. We can observe the difference that RANSAC makes when implemented, making the estimation more accurate and reliable.



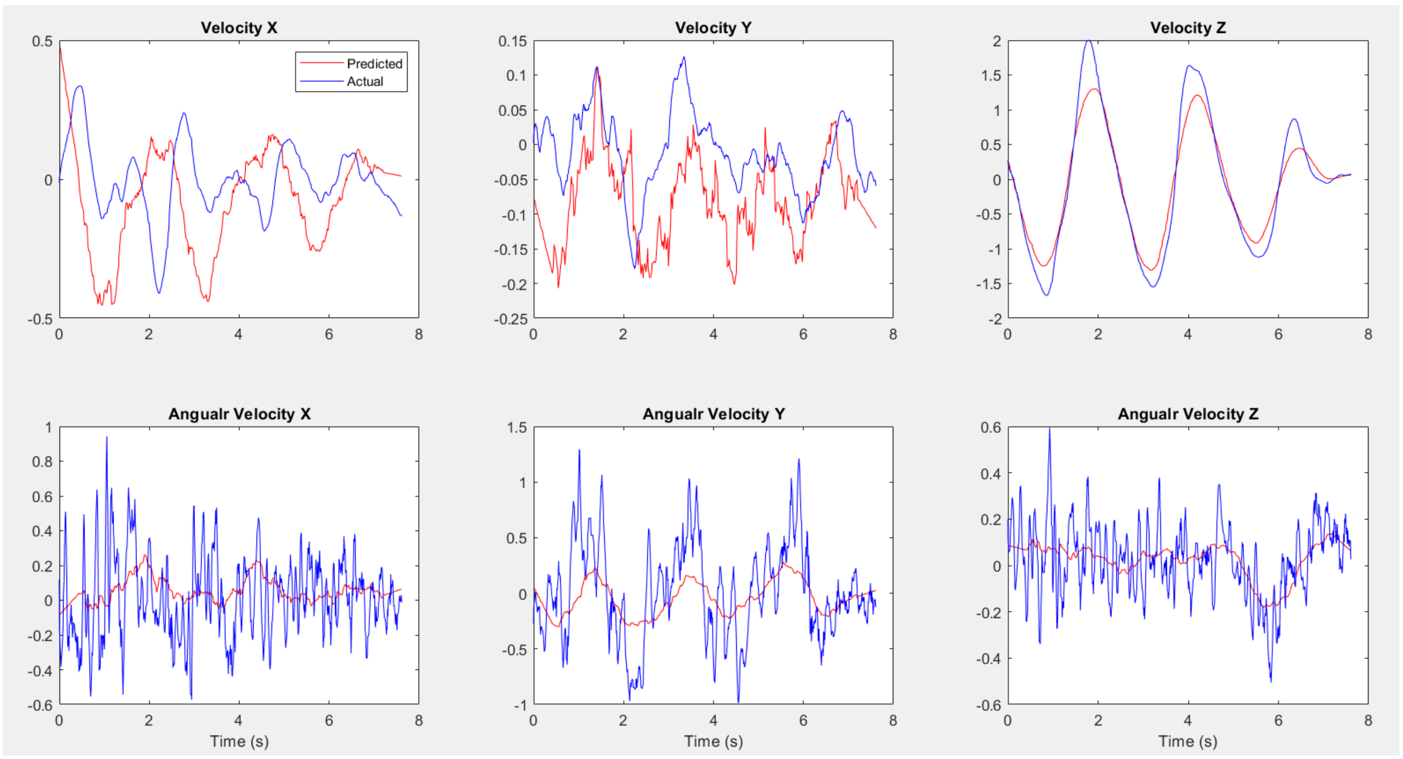
P-2: Outcomes of Optical Flow for Dataset-1 w/o RANSAC



P-2: Outcomes of Optical Flow for Dataset-1 with RANSAC



P-2: Outcomes of Optical Flow for Dataset-4 w/o RANSAC



P-2: Outcomes of Optical Flow for Dataset-4 with RANSAC

CONCLUSION:

The position and orientation estimation of the drone though the use of April tag and further calculating the velocities of the drone travelling through a space has proved to be an effective methodology without relying on expensive external tracking systems which makes the system more independent of external devices for its functionality.

The use of optical flow to calculate the velocity of the system and the position and orientation obtained can be used for the attitude control of the drone thus making this approach reliable. The outcomes of the implementation of RANSAC has proved to be effective in removing most outliers and refining the data further for better optimisation.