Sujay Nayak Anil
*ECE Department*
*NYU Tandon School of Engineering*
*N10462423 (san440)*

## INTRODUCTION

The aim of this report is to demonstrate the implementation of pose estimation of a quadrotor using an onboard camera and identifiable markers placed in known coordinates in the world frame and also to demonstrate the implementation of velocity estimation of the quadrotor using feature tracking between the consecutive frames captured by the camera. A process model is designed to estimate the pose of the robot with the help of the sensory data received from the camera and the prior knowledge of the pose and position of the identifiable markers in the world frame. For the simplicity of implementation, the pinhole model of the camera is considered for this implementation.

## IMPLEMENTATION

The intrinsic matrix of the camera used is given to be :

```
K =

  311.0520         0  201.8724
         0  311.3885  113.6210
         0         0    1.0000
```

## Part A. Pose estimation

The relation between a point in the world frame and its corresponding point in the image frame of the camera is given by :

$$\lambda \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = K \left( {}^{c}R_{w} \;\; {}^{c}T_{w} \right) \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix}$$

In the scenario presented in the project, the identifiable markers are placed on the ground and coordinates of the corners of the markers w.r.t to the world frame are known.

At $Z_w = 0$, we have

$$\underbrace{\begin{pmatrix} x \\ y \\ 1 \end{pmatrix}}_{\text{Pixel Coordinates}} \sim \underbrace{\begin{pmatrix} f & 0 & x_0 \\ 0 & f & y_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & t_1 \\ r_{21} & r_{22} & t_2 \\ r_{31} & r_{32} & t_3 \end{pmatrix}}_{\substack{\text{Projective Transformation} \\ \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix}}} \underbrace{\begin{pmatrix} X_w \\ Y_w \\ 1 \end{pmatrix}}_{\text{Planar Coordinates}}$$

Above equation is of form :
$\lambda \,.\, x = H \,.\, X$
where $\lambda$ - scalar quantity



Email : *san440@nyu.edu*
Uploaded on Apr 9th 2020

We know that,
S(x) . H . X = 0 holds true for such equation
where S(x) - Skew-symmetric matrix of vector x

Simplifying the equation will get us :

$$\begin{pmatrix} x_i & y_i & 1 & 0 & 0 & 0 & -x'_i x_i & -x'_i y_i & -x'_i \\ 0 & 0 & 0 & x_i & y_i & 1 & -y'_i x_i & -y'_i y_i & -y'_i \end{pmatrix} \underbrace{\boxed{h}}_{\text{Vector of unknown transformation parameters}} = 0$$

where x , y - coordinates of the point in the world frame
x' , y' - corresponding coordinates in the image frame of the camera
h - H matrix rewritten as a vertical vector of length 9

In order to solve the above equation for the 9 unknown parameters, 9 equations are needed. Since the projective transformation is scale independent, only 8 equations are required to solve for the unknown transformation parameters. Here each point from the world frame can provide 2 equations, so we would need at least 4 points to solve for the unknown parameters.

In the process model, the camera onboard of the quadrator captures the scene of the world and tries to identify any markers present in the frame captured. Once the markers are identified, the corresponding corners of the markers are identified in the image frame and the information of the point in the world frame is retrieved from the prior Knowledge.

Note : Each marker is unique and has unique coordinates in the world frame.

The equation for solving the unknown transformation parameters can be rewritten as :

$$Ah = 0$$

The equation is solved by performing single value decomposition on the A matrix.

Once the H matrix is calculated, we need to extract the rotation matrix and the translation vector from the H matrix. Before we do that, we need to decompose the H matrix into camera intrinsic matrix and altered homogeneous matrix. The altered homogeneous matrix is obtained by multiplying inv(K) to the obtained H matrix.

$$\begin{pmatrix} \hat{r}_{11} & \hat{r}_{12} & \hat{t}_1 \\ \hat{r}_{21} & \hat{r}_{22} & \hat{t}_2 \\ \hat{r}_{31} & \hat{r}_{32} & \hat{t}_3 \end{pmatrix} = \begin{pmatrix} f & 0 & x_0 \\ 0 & f & y_0 \\ 0 & 0 & 1 \end{pmatrix}^{-1} \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix}$$

From the information of the first and second column of the altered rotation matrix, the third column of the rotation matrix is reconstructed to fit the underlying principles of a rotation matrix.

$$(\hat{R}_1 \ \ \hat{R}_2 \ \ \hat{R}_1 \times \hat{R}_2) = USV^T$$

$$R = U \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \det(UV^T) \end{pmatrix} V^T$$



Email : *san440@nyu.edu*

The extracted translation vector is scaled appropriately to obtain the original translation vector.

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

We have to note that the rotation matrix obtained denotes the rotation of the world frame w.r.t the camera frame and translation vector denotes the translation of the world frame w.r.t the camera frame. In order to obtain rotation and translation of the quadrotor w.r.t the world frame, appropriate transformations are applied to the obtained homogeneous transform matrix.

We know,

$H^{w}_{c} \cdot H^{c}_{r} = H^{w}_{r}$

$H^{w}_{r} = (H^{c}_{w})^{-1} \cdot H^{c}_{r}$

where ,
$H^{w}_{r}$ - transformation of quadrotor pose w.r.t the world frame
$H^{c}_{w}$ - Obtained transformation between onboard camera pose and the world frame
$H^{c}_{r}$ - known transformation between quadrotor pose and the onboard camera pose

Hence, the pose and position of the quadrotor was determined w.r.t the world frame.

## Part B. Velocity estimation

In order to estimate the velocity of the quadrotor w.r.t the world, we would need the sensory data from the onboard camera for two consecutive frames of the capturing scene of the world. The theory behind estimating the velocity of the camera w.r.t the world frame is that :
If one can precisely track how points in one image frame have shifted compared to its previous frame, then it is possible to estimate the relative velocity of the camera w.r.t the world provided the pose and position of the camera is known w.r.t the world frame. Here, we make use of the pose estimation from our previous process model to aid us in the velocity estimation.

For tracking the points between the consecutive frames, the in-built computer vision toolbox from MATLAB was used. At the beginning, the strongest features were identified from the previous image frame. The following points of the corresponding features were tracked to the current image frame. The optical flow for each corresponding points is calculated using the formulae :

$$optical\ flow = \frac{Current_{Pixel\ Location} - Previous_{Pixel\ Location}}{dt}$$

where,
dt - captured time difference between the consecutive image frames



Email : *san440@nyu.edu*

Once the optical flow is determined, the velocity of the camera w.r.t the world frame can be determined using the formulae:

$$\frac{1}{Z}\underbrace{(\mathbf{p}e_3^T - I)}_{\mathbf{A}(\mathbf{p})}\mathbf{V} + \underbrace{(I - \mathbf{p}e_3^T)[\mathbf{p}]_\times}_{\mathbf{B}(\mathbf{p})}\mathbf{\Omega}$$

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

where,
Z - depth of the point in the world w.r.t the camera frame
p - coordinate of the point in the image frame
p' - optical flow of the point in the image frame
V - Positional velocity of the point in the camera frame
Ω - Angular velocity of the point in the camera frame coordinates and optical flow in the

The above equation is determined for normalised coordinates and optical flow in the image frame. For standard coordinates and optical flow in the image frame, we can use :

$$p' = \frac{1}{Z}.(p.e3^{\,T} - K)V + (K - p.e3^{\,T})S(K^{-1}.p)\Omega$$

Though mathematically both can yield the same result, it is found that computationally it is more efficient to normalize the coordinates and then calculate the optical flow using the previous formulae.

The relation can be rewritten as :

$$\begin{pmatrix} u \\ v \end{pmatrix} = \frac{1}{Z}\begin{pmatrix} -1 & 0 & x \\ 0 & -1 & y \end{pmatrix}\begin{pmatrix} V_x \\ V_y \\ V_z \end{pmatrix} + \begin{pmatrix} xy & -(1+x^2) & y \\ 1+y^2 & -xy & -x \end{pmatrix}\begin{pmatrix} \Omega_x \\ \Omega_y \\ \Omega_z \end{pmatrix}$$

where,
x - coordinate of the point in x-axis of the image frame
y - coordinate of the point in y-axis of the image frame
u - Optical flow of the point in x-axis of the image frame
v - Optical flow of the point in x-axis of the image frame

Similar to the previous process flow, we need to solve the equation for 6 unknown variables. Here optical flow from each point can provide 2 equations, so we would need at least 3 points to solve for the unknown variables. In the following implementation, 4 points were used to solve the equation for better results.

In the process model, RANSAC is implemented for velocity estimation at each frame to choose the best possible velocity for the given scenario. At each frame, velocity is estimated using 4 randomly chosen points from the set of traced points for *k,* number of iterations. For every velocity calculated, the number of inliners are calculated based on the predefined threshold. The velocity with maximum number of inliners is selected as the best fit for the given scenario.

Threshold selected for the implementation is *0.01* .
The *k*, number of iterations is defined by :

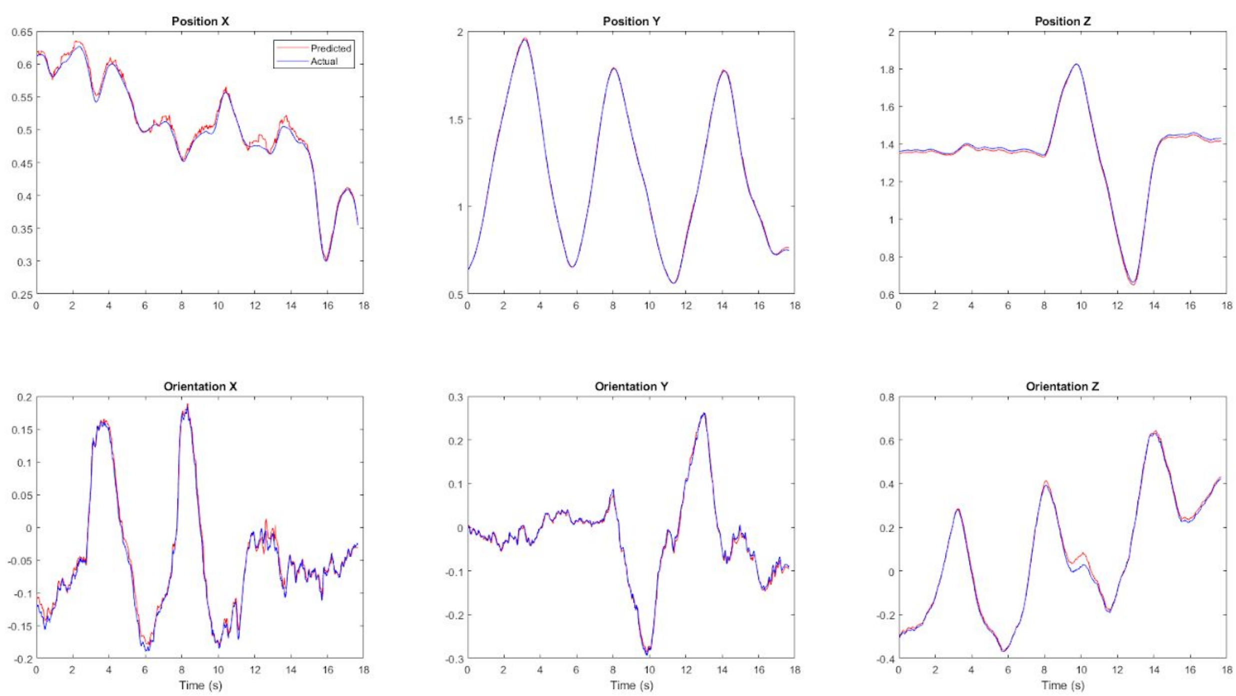
$$K = \frac{log(1-p)}{log(1-e^{M})}$$

where,
p - probability of desired success
(0.99 selected)
e - probability of matching points
(0.5 selected)
M - Number of points chosen at a time
(3 is this case)

The velocity obtained from the estimation is the velocity of the camera w.r.t the world frame represented in the camera frame. Necessary transformation is applied to estimate the velocity of the quadrotor w.r.t the world frame.
Since $\Omega^{c}_{quad} = \Omega^{c}_{camera}$,
$\Omega^{w}_{quad} = R^{w}_{c}.\Omega^{c}_{quad}$
$Vel^{w}_{quad} = R^{w}_{c}\ (\ Vel^{c}_{camera} + S(\Omega^{c}_{camera}).\ T^{c}_{c\text{ - }quad}\ )$

Hence, the positional and angular velocity of the quadrotor was determined w.r.t the world frame.

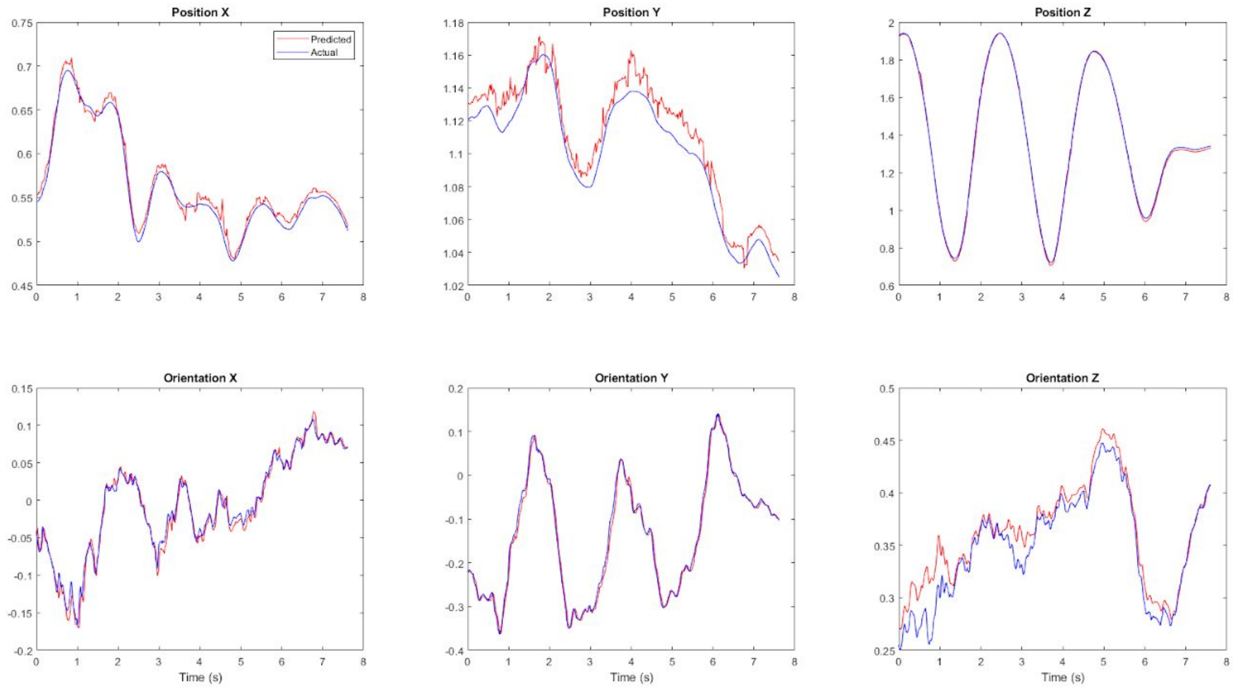
Plot for *pose estimation* for Dataset 1 :



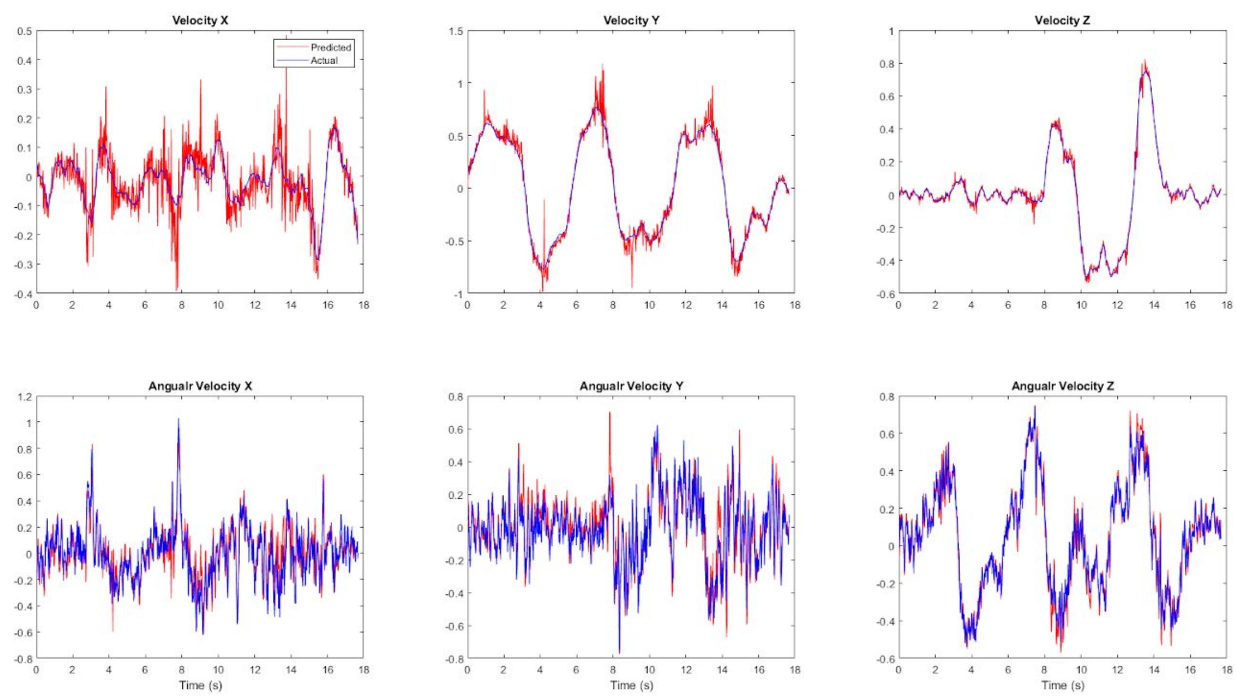



Email : *san440@nyu.edu*
Uploaded on Apr 9th 2020

Plot for *pose estimation* for Dataset 4 :



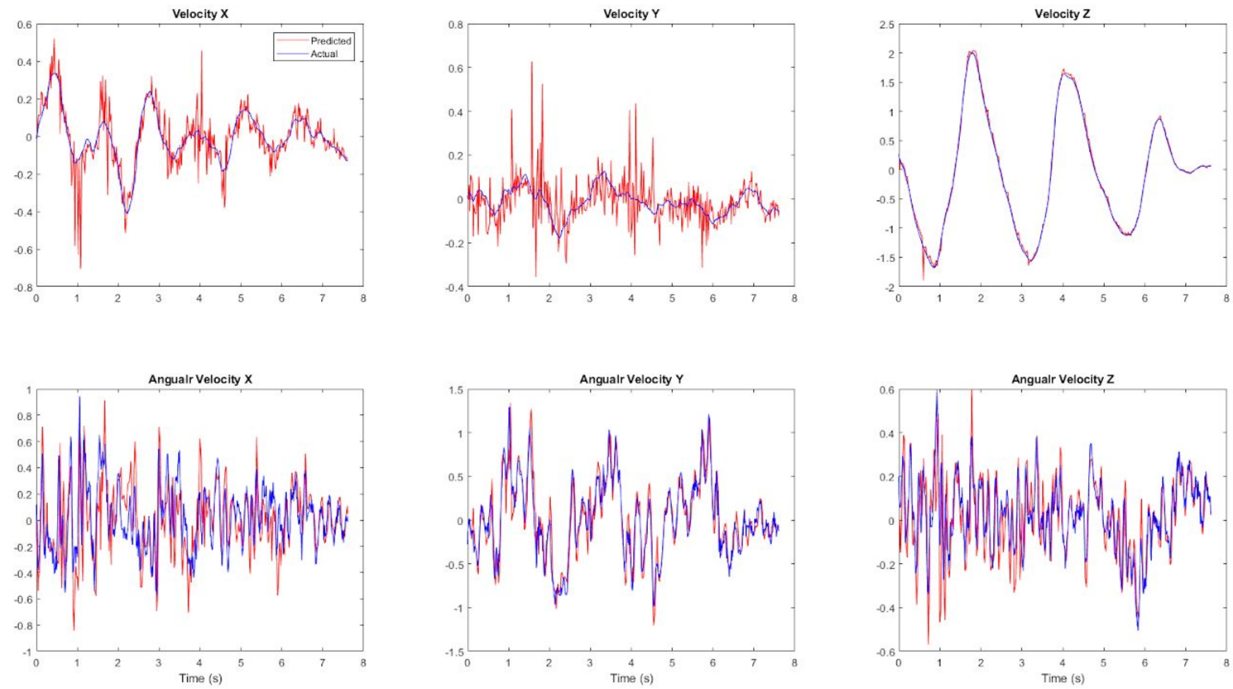

Plot for *velocity estimation* for Dataset 1 :






Email : *san440@nyu.edu*

Plot for *velocity estimation* for Dataset 4 :




## Conclusion

For part - I , we can observe that the estimated pose and position of the quadrotor is almost ideally following the ground truth of the position and orientation of the quadrotor w.r.t the world frame determined by the vicon system.

For part - II , we can observe that the estimated positional and angular velocity of the quadrotor is adequately following the ground truth of the positional and angular of the quadrotor w.r.t the world frame determined by the vicon system. We can also observe the presence of noise in the velocity estimation of the quadrotor. This is due to the fact that points detected in the previous frame are not precisely traced to the corresponding points in the current frame and also the velocity estimation model assumes that the camera is translating parallel to the world frame, which in reality is not true.



Email : *san440@nyu.edu*