Report

Project 2

Submitted By: Krishnesh Meratia

Abstract:

A vision based 3-D pose estimator which estimates position and orientation in part 1 and linear and angular velocity estimator in part 2 of the quadrotor based is implemented. The project has been divided in two phases in first pose is estimated using AprilTags, and in second a vision based features extraction algorithm is used is used to estimate the velocities.

Background:

A pin-hole camera model is:

$$\lambda \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} f & 0 & x_0 \\ 0 & f & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R_w^c & T_w^c \end{bmatrix} \begin{pmatrix} x_w \\ y_w \\ z_w \\ 1 \end{pmatrix} \quad - (1)$$

$\lambda =$ scaling factor due to projective transformation

$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} =$ pixel coordinates of projected point

$\begin{bmatrix} f & 0 & x_0 \\ 0 & f & y_0 \\ 0 & 0 & 1 \end{bmatrix} =$ camera intrisic matrix has the focal length of camera and paramteres to correct distortion found using the checker-board method were given to us.

$\begin{bmatrix} R_w^c & T_w^c \end{bmatrix} =$ rotation matrix from camera to world and postion of the world point wrt to camera

$\begin{pmatrix} x_w \\ y_w \\ z_w \\ 1 \end{pmatrix} =$ position of the point in the world frame

Part 1)

Our aim is to find $\begin{bmatrix} R_w^c & T_w^c \end{bmatrix}$ we have been given a mat of Apriltags (which has four points p1, p2, p3 and p4) for every tag we found the coordinates of these points in the world hence we have $\begin{pmatrix} x_w \\ y_w \\ z_w \\ 1 \end{pmatrix}$ as the mat is on the floor z_w is zero stored in graph.mat file

We know the coordinates of these points in the image frame $\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix}$

We can write this equation(1) in form of projection transformation:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} C & t \\ v & b \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

$\begin{bmatrix} C & t \\ v & b \end{bmatrix} =$ projective transformation or homography (H)

It can be determined up to only a scale factor λ as A and λA represent the same transformation hence of the 9 elements we can find only 8 elements.

$$\lambda\begin{pmatrix} x'_i \\ y'_i \\ 1 \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}\begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix}$$

The eight DOF can be found using linear algebra

$$\begin{bmatrix} x_i & y_i & 1 & 0 & 0 & 0 & -x'_i x_i & -x'_i y_i & -x'_i \\ 0 & 0 & 0 & x_i & y_i & 1 & -y'_i x_i & -y'_i y_i & -y'_i \end{bmatrix} h = 0 \quad Ah = 0$$

which gives us 2 equations we need 8 equation to solve 8 DOF each point gives us 2 equation we need 4 points.

h = vector of unknown transformation parameters $(h_{11} \ldots\ldots h_{33})$

Is solved using singular value decomposition

A = USV^T

h = 9th column of the V

We get multiple Apriltags at a time we can make a big A matrix to solve for H.

Once we have found the H (homography transform)

Camera model as projective transform

$$\lambda\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} f & 0 & x_0 \\ 0 & f & y_0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix}\begin{pmatrix} x_w \\ y_w \\ z_w \\ 1 \end{pmatrix}$$ as z_w is 0 as mat is on ground the third column of the matrix $\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix}$ can be removed which gives us the equation

$$\lambda\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} f & 0 & x_0 \\ 0 & f & y_0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} r_{11} & r_{12} & t_1 \\ r_{21} & r_{22} & t_2 \\ r_{31} & r_{32} & t_3 \end{bmatrix}\begin{pmatrix} x_w \\ y_w \\ 1 \end{pmatrix}$$ comparing with projective equation

$$\begin{bmatrix} f & 0 & x_0 \\ 0 & f & y_0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} r_{11} & r_{12} & t_1 \\ r_{21} & r_{22} & t_2 \\ r_{31} & r_{32} & t_3 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \qquad \begin{bmatrix} f & 0 & x_0 \\ 0 & f & y_0 \\ 0 & 0 & 1 \end{bmatrix} = K$$

$$\begin{bmatrix} r_{11} & r_{12} & t_1 \\ r_{21} & r_{22} & t_2 \\ r_{31} & r_{32} & t_3 \end{bmatrix} = K^{-1}\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} = (\hat{R}_1 \quad \hat{R}_2 \quad \hat{T})$$

The rotation matric is solved satisfying constraints as

$(\hat{R}_1 \quad \hat{R}_2 \quad \hat{R}_1\hat{R}_2)$ = USV^T singular value decomposition

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

R = rotation matrix is used for solving the orientation

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

This gives us the relation between the camera frame and world frame we need relation between world frame and robot frame which can be solved as:

We have found $H_w^c = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}$

$$H_c^w = \begin{bmatrix} R' & -R'T \\ 0 & 1 \end{bmatrix}$$

We are given $H_r^c = \begin{bmatrix} 0.7071 & -0.7071 & 0 & -0.04 \\ -0.7071 & -0.7071 & 0 & 0 \\ 0 & 0 & -1 & -.03 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

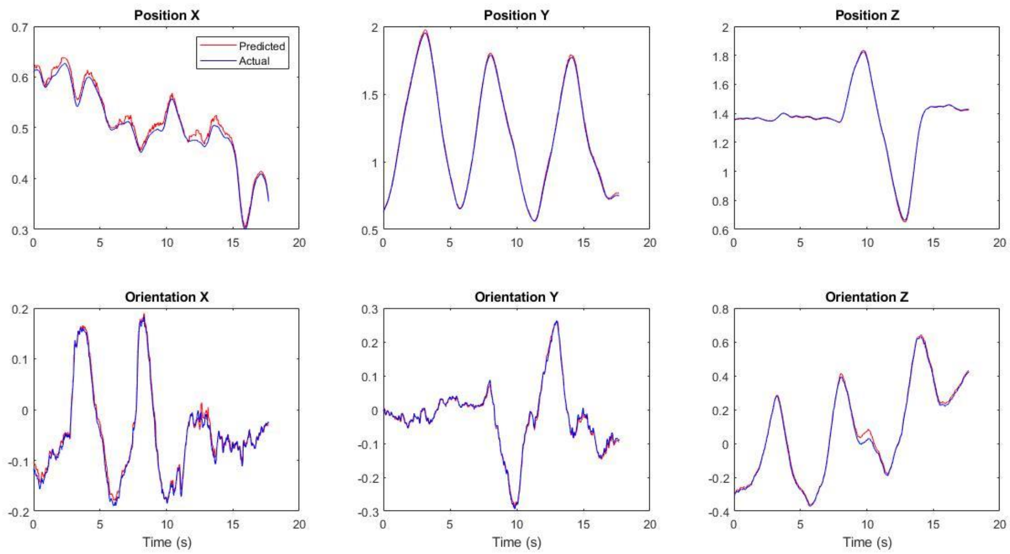
$H_r^w = H_c^w \, H_r^c = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

$t_1, t_2, t_3 =$ position in x, y and z respectively

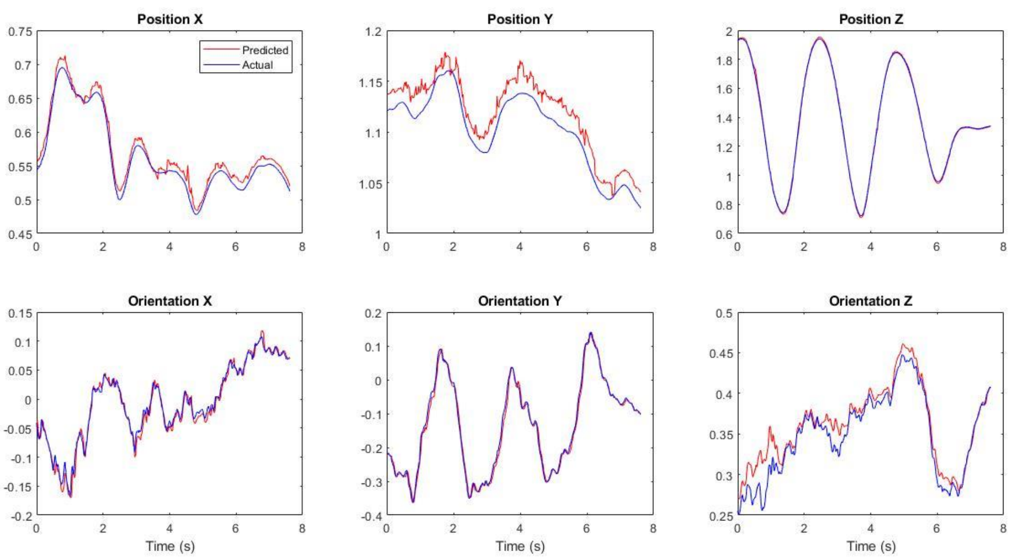
$\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$ is used for euler XYZ orientation

Results:

Data 1



Data 4



Part 2

A computer- vision based feature extraction and features tracking algorithm is used.

Good features are selected based on the eigen values of the equation

$$\sum_{(x,y)} \nabla I_{t+1}(x,y)\, \nabla I_{t+1}(x,y)^T$$

Which uses the correspondence between the two images at time t and t+1

$I_t(x,y)$ and $I_{t+1}(x+\delta x, y+\delta y)$

$$\begin{bmatrix} \delta x \\ \delta y \end{bmatrix} = \mathrm{inv}\left(\sum\nolimits_{(x,y)} \nabla I_{t+1}(x,y)\, \nabla I_{t+1}(x,y)^T\right) \sum_{(x,y)} \nabla I_t(x,y)\, \nabla I_{t+1}(x,y)$$

Feature extraction is done for a window once the features are extracted these features are tracked in the second image and we know how much the pixels has moved this displacement is formulated to find the velocity of the camera wrt the world expressed in the camera frame.

$V = \begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix}$ Translation velocity of the camera wrt world in the camera frame

$\omega = \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix}$ Angular velocity of the camera wrt world in the camera frame

x, y are pixel coordinates given by the algorithm in in image frame to bring these in the camera frame we multiply by inverse of camera intrinsic matrix

$xc = \mathrm{inv}(K)x$ $\qquad \delta xc = xc(t+1) - xc(t)$

$yc = \mathrm{inv}(K)y$ $\qquad \delta yc = yc(t+1) - yc(t)$

$\delta xc =$ X displacement in the camera frame

$\delta yc =$ Y displacement in the camera frame

$u = \delta xc/dt$ velocity X of pixels in the camera frame

$v = \delta yc/dt$ velocity Y of pixels in the camera frame

$$\begin{pmatrix} u \\ v \end{pmatrix} = \frac{1}{z}\begin{pmatrix} -1 & 0 & xc \\ 0 & -1 & yc \end{pmatrix}\begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix} + \begin{pmatrix} xc\,yc & -(1+xc^2) & yc \\ 1+yc^2 & -xc\,yc & -xc \end{pmatrix}\begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \text{—(a)}$$

Z is calculated using the pin model camera equation

$$Z\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} f & 0 & x_0 \\ 0 & f & y_0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix}\begin{pmatrix} x_w \\ y_w \\ z_w \\ 1 \end{pmatrix}$$

$\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix}$ found in part 1

As $z_w = 0$

$$Z\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} f & 0 & x_0 \\ 0 & f & y_0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} r_{11} & r_{12} & t_1 \\ r_{21} & r_{22} & t_2 \\ r_{31} & r_{32} & t_3 \end{bmatrix}\begin{pmatrix} x_w \\ y_w \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} xc \\ yc \\ 1 \end{pmatrix} = \mathrm{inv}\left(\begin{bmatrix} f & 0 & x_0 \\ 0 & f & y_0 \\ 0 & 0 & 1 \end{bmatrix}\right)\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix}$$

$$Z\begin{pmatrix} xc \\ yc \\ 1 \end{pmatrix}=\begin{bmatrix} r_{11} & r_{12} & t_1 \\ r_{21} & r_{22} & t_2 \\ r_{31} & r_{32} & t_3 \end{bmatrix}\begin{pmatrix} x_w \\ y_w \\ 1 \end{pmatrix}$$ done when bring the pixel coordinate from image frame to camera frame

$$\mathrm{Inv}\left(\begin{bmatrix} r_{11} & r_{12} & t_1 \\ r_{21} & r_{22} & t_2 \\ r_{31} & r_{32} & t_3 \end{bmatrix}\right)\begin{pmatrix} xc \\ yc \\ 1 \end{pmatrix} = \begin{pmatrix} x_w/z \\ y_{w/z} \\ 1/z \end{pmatrix}$$

We have LHS

Z is the 1/ (3,1) element of the LHS after solving

Equation (a) has 6 unknown and 2 equations so require at least 3 points to solve it.

It has been done using two methods

1) Least square
2) Ransac

Least square

At time t we choose 50 good feature points and applied least square

Rewrote eq(a) in matrix form as

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{bmatrix} -1 & 0 & xc & xc\,yc & -(1+xc^2) & yc \\ 0 & -1 & yc & 1+yc^2 & -xc\,yc & -xc \end{bmatrix}\begin{pmatrix} v_x \\ v_y \\ v_z \\ \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} == (AX = B)$$

Formed a big matrix A and B for all the feature extracted points

Solved for least square:

$X = (A^TA)^{-1}A^TB$ $\quad$ $V = X(1:3)$ $\quad$ $\omega = X(4:6)$ $\quad$ rotation needs to be performed to change the frame

Ransac:

Is a general parameter estimation approach designed to cope with a large proportion of outliers in the input data.

We assume that 60% of our data is good with a tolerance of 0.01

e = 0.6 -threshold

We repeat for k iteration and hit a inlier set with a probability of success psuccess in k iteration

psuccess = 0.95

k is calculated using

$k = \log(1\text{-psuccess})/\log(1- e^m)$

m = min no of points required to determine the model parameters = (3)

Procedure followed (from)

http://www.cse.yorku.ca/~kosta/CompVis_Notes/ransac.pdf

1)Randomly choose 3 points and calculated the model parameters for those 3 points.

2)Determined how many points from set of all points fit with the tolerance.

3)If the fraction of the number of inliers over the total number points in the set exceeded a predefined threshold e, re-estimated the model parameters using all the identified inliers and terminated.

4)Otherwise, repeated the step 1-3 for a maximum of k times

5) if no inlier set with threshold was hit model parameters were calculated using the set with max no of inliers.

Ransac fixes the problem with the number of iterations growing too fast.

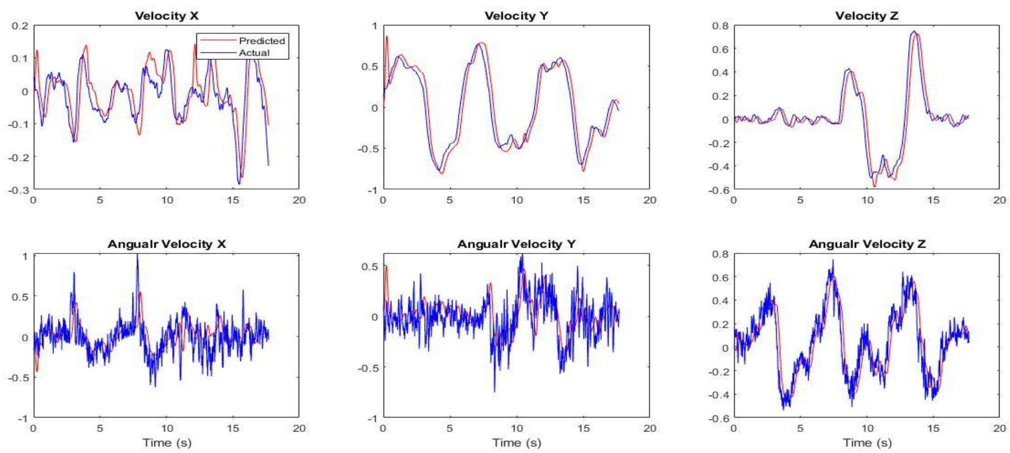
This gave us X $\quad V = X(1:3) \quad \omega = X(4:6)$

We have the velocities in the camera frame wrt world expressed in camera frame
We need velocities of the robot wrt to the world:

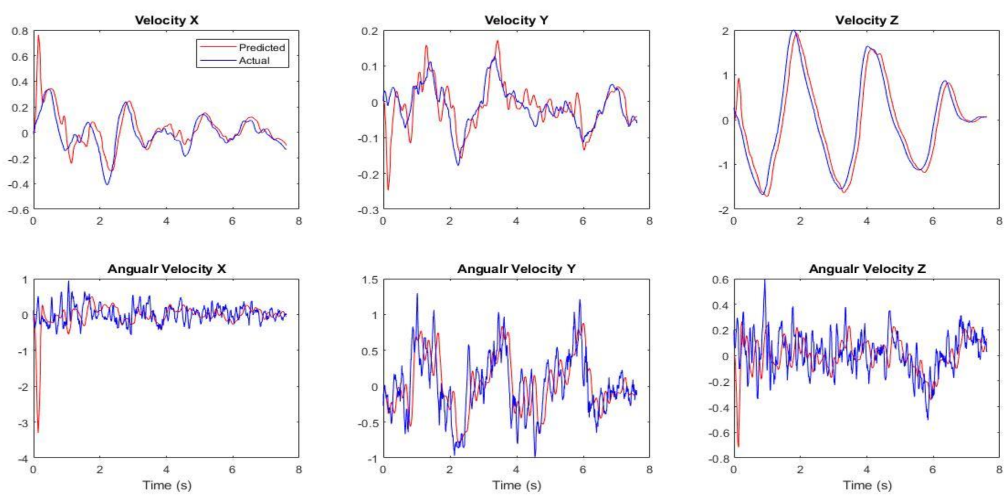
$$\begin{bmatrix} \dot{p}_r^r \\ \omega_r^r \end{bmatrix} = \begin{bmatrix} R_c^r & -R_c^r S(p_r^c) \\ 0 & R_c^r \end{bmatrix} \begin{bmatrix} \dot{p}_c^c \\ \omega_c^c \end{bmatrix}$$

$$\dot{p}_r^w = R_r^w \dot{p}_r^r \qquad \omega_r^w = R_r^w \omega_r^r$$

Fir1 low pass filter was used on time and velocities, cutoff frequencies were found.
Results
Least square results:
Data 1:

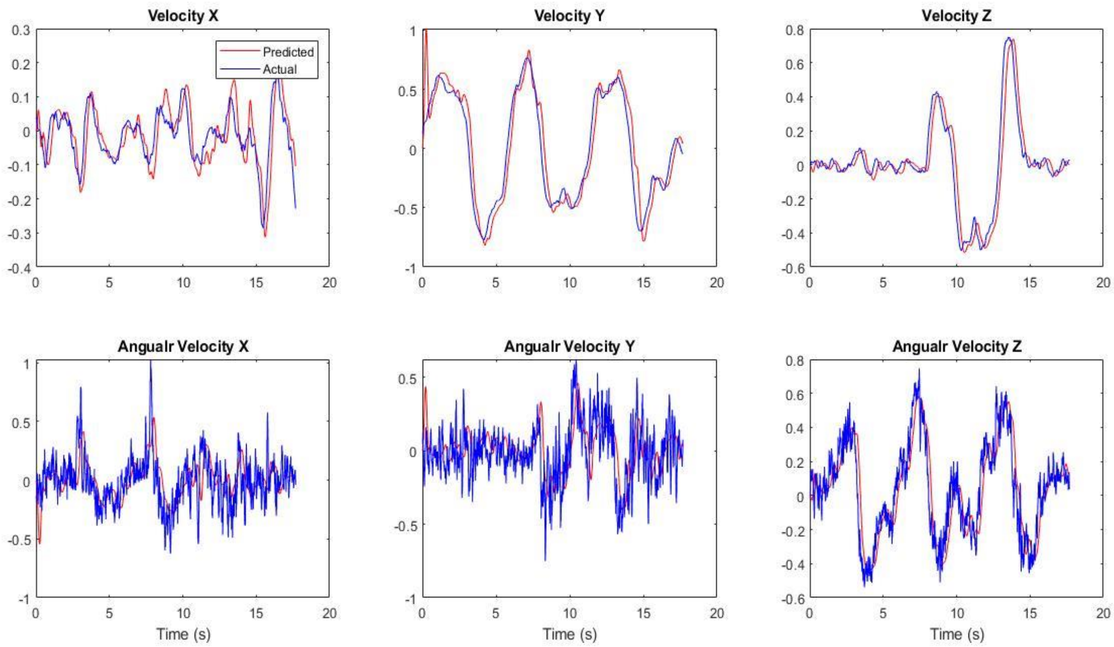


Data 2

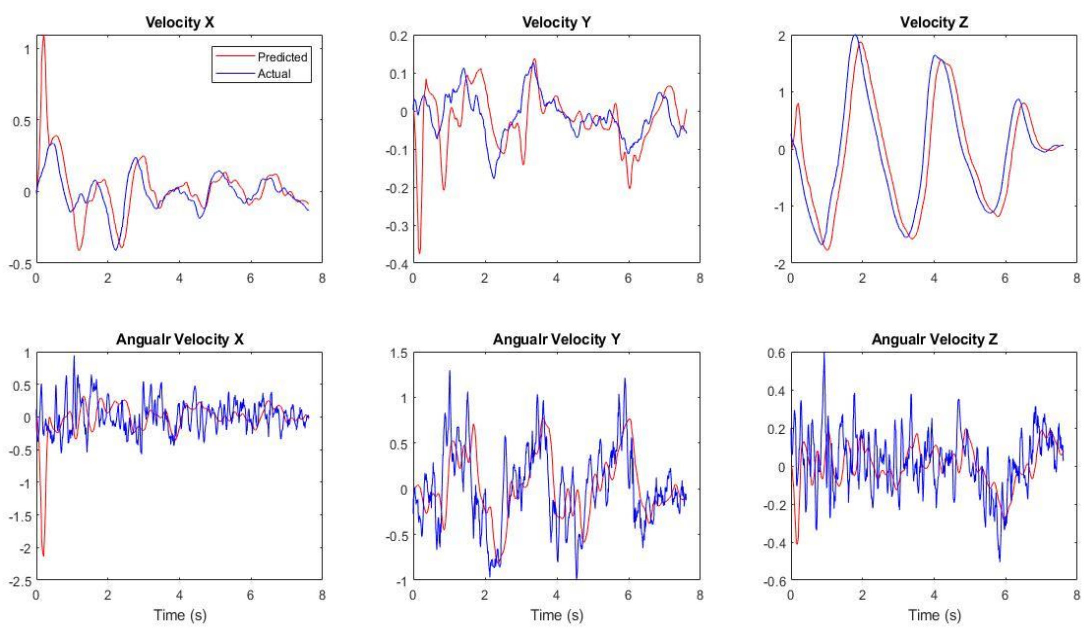


Ransac:

Data 1



Data 4



Initial peaks were removed if no time filter was used.