ROB-6213: Project 2

Name: Puda Zhao
studentID: N13625214
netID: pz2078

Part 1

Pose Estimation

This part is in '1_pose_estimation':

Step1. Read and calculate coordinates of points in both camera frame and world frame.

The coordinates of points in camera frame in pixel is saved in date provided. We could read them directly.

The coordinates of points in world frame can be computed by the AprilTag mat. For every QR code, we could firstly compute the coordinate of corner p4. Then from p4 we could get all of the corners.

The details of the method are showed in file 'getCorner.m'.

Step2. Solve the projective transformation equation

We need to solve matrix H from the equation below:

$$\begin{pmatrix} x_c \\ y_c \\ 1 \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{32} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{pmatrix} \begin{pmatrix} x_w \\ y_w \\ 1 \end{pmatrix}$$

From step1, we have already got p_c and p_w. With more than 4 groups of p_c and p_w, it is enough for us to solve this equation.

Rewrite the equation like:

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

With n ID showed in 1 image, we got $4n$ groups of points. Which means A will be a $8n$*9 Matrix.

Then we could use SVD to solve H from the equation. Align the sign of the h matrix after the first SVD.

Step3. Pose from Projective Transformations

$$(\hat{R}_1 \quad \hat{R}_2 \quad \hat{T}) = K^{-1}H$$

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

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

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

$H = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{32} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{pmatrix}$ results from step2, $K = \begin{pmatrix} 311.0520 & 0 & 201.8724 \\ 0 & 311.3885 & 113.6210 \\ 0 & 0 & 1 \end{pmatrix}$

Step4. Transform frame

The final output should be in body frame to world frame.

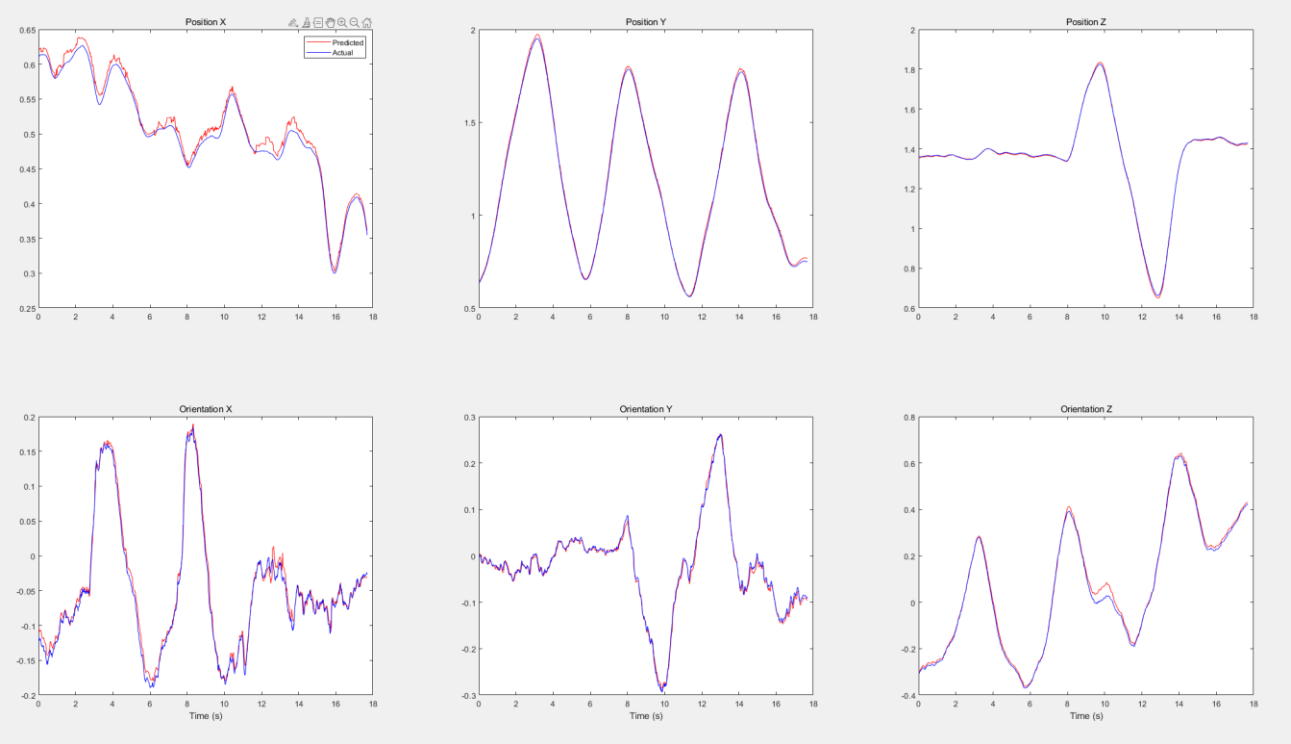
We could get T_c^w from step3, and T_c^i from the given parameters of the quadrotor.

Detailed methods are in files.

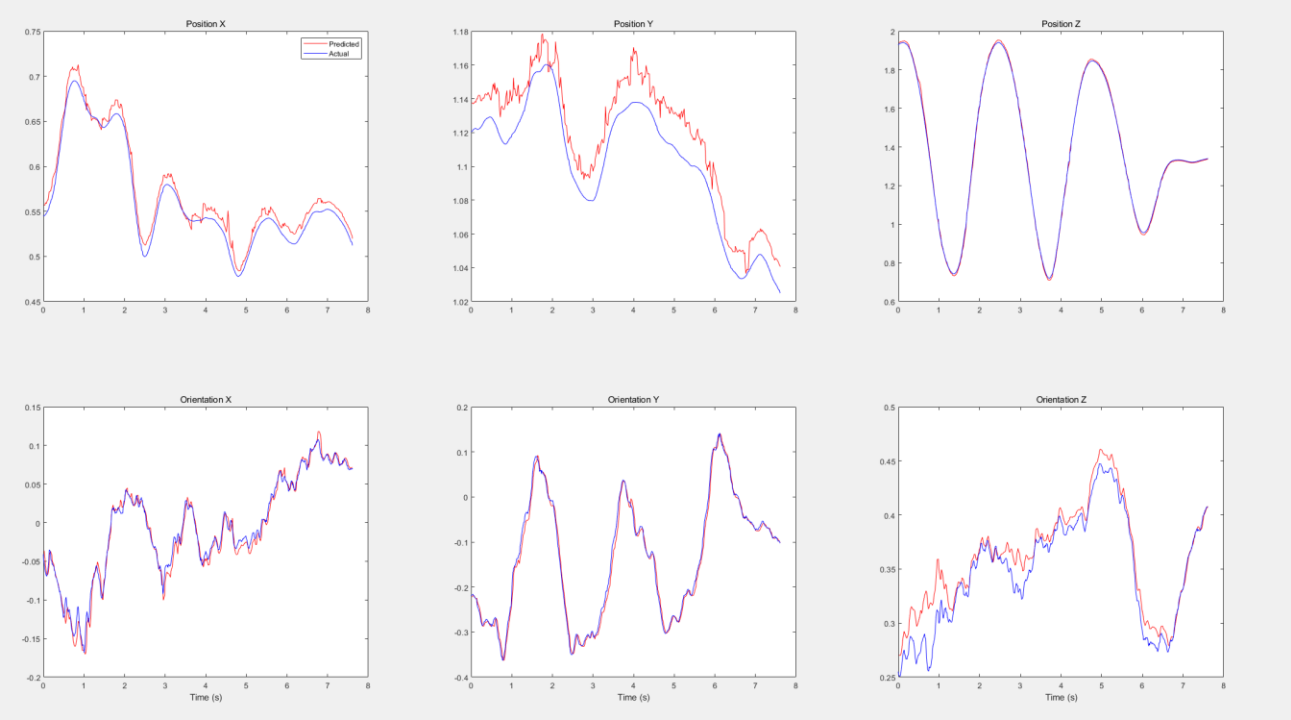
$$T_b^w = T_c^w T_c^i$$

Plotting results:

Datasheet 1:



Datasheet 4:



Part 2

Optical Flow

This part is in '2_optical_flow':

Step1. Find good features and track them.

We could find useful function in Computer Vision Toolbox.

To find features of the image. We could use 'detectFASTFeatures' to get the coordinates of the corners you want to track.

Then we could use 'vision.pointTracker' to track them in the next frame. The function will return coordinates of the corners that you are tracking in the next frame.

Step2. Prepare for the Motion Field Equations

1. Height: Z is the height of the drone which comes from the 'estimatePose'. Also, we need to transfer Z_w to Z_c
2. p and $\dot{p}$: We could get them from step1, $\dot{p} = \Delta p / \Delta t$
3. Use cameraMatrix to transform the units of p and $\dot{p}$ from pixel to meters.

Step3. Solve Motion Field Equations

By the information from the steps below, we could now try solving the motion field equations:

With:

$$e_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

$$A(p) = pe_3^T - I$$

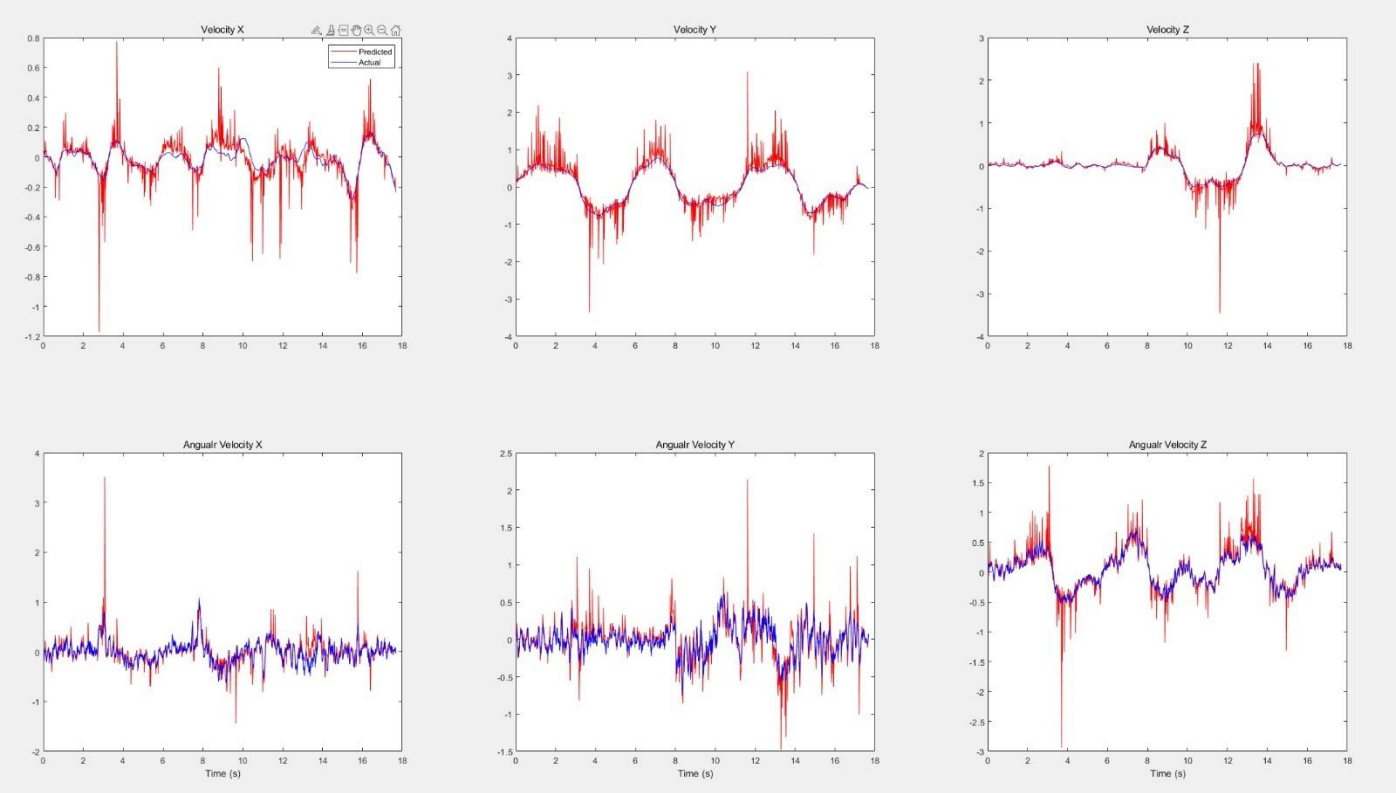
$$B(p) = (I - pe_3^T) * \hat{p}$$

$$H_i(p) = \begin{bmatrix} \frac{1}{Z} A(p) & B(p) \end{bmatrix}$$

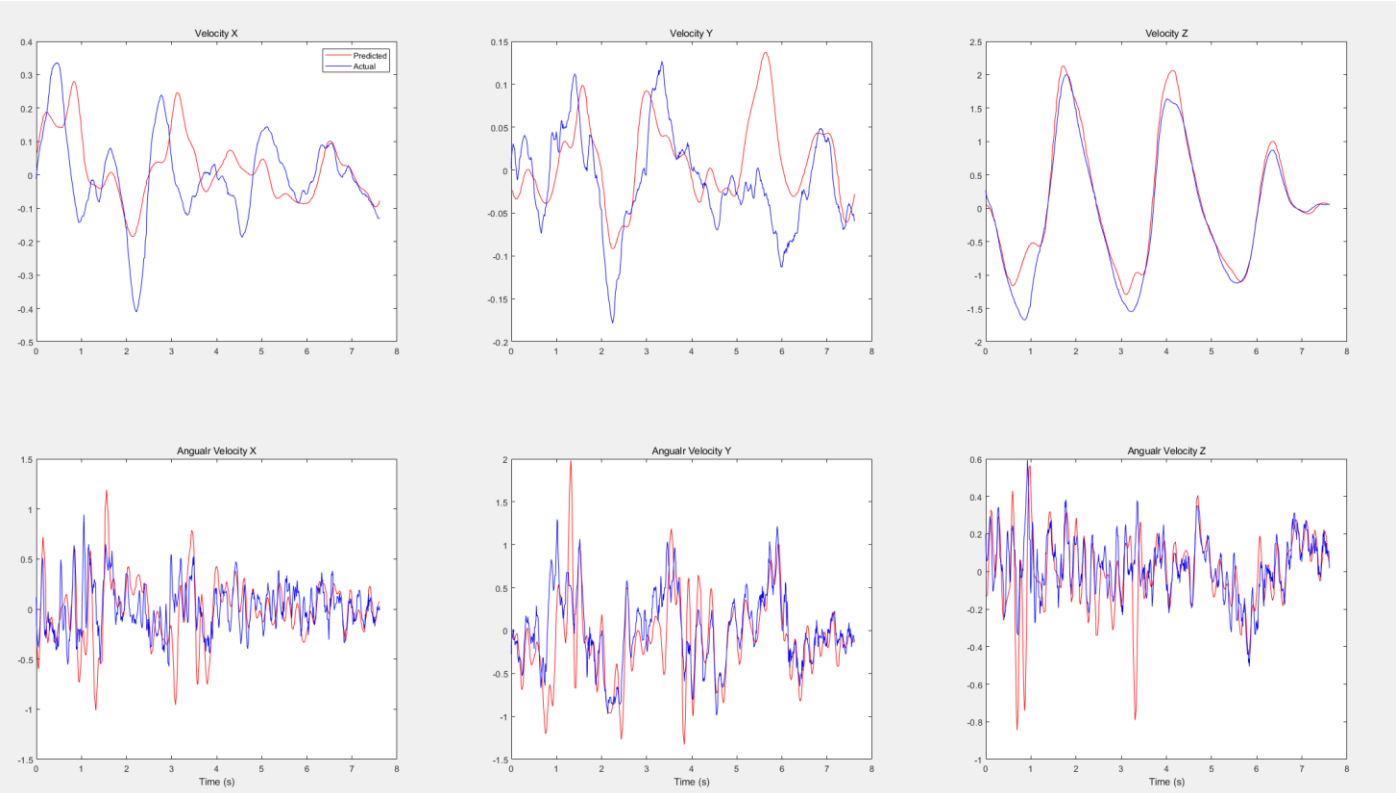
$$\begin{pmatrix} V^* \\ \Omega^* \end{pmatrix} = H^+ \dot{p}$$

Plotting results **without RANSAC:**

Datasheet 1:



Datasheet 4:



RANSAC

This part is in 'velocityRANSAC.m':

Step1. Choose a minimal sample set

In this case, we need to select at least 3 groups of p and $\dot{p}$ to solve the motion field equation

Step2. Count the inliers for this set

The core problem in this step: How to judge a point whether it is an inlier?

For 2D linear model, we often use the distance between the sample point and the linear model to decide whether the sample point is an inlier.

$$dis = \frac{|Ax + b|}{\|A\|}$$

For this model, we could also use this method as:

$$d = \frac{|HX - \dot{p}|}{\|X\|}$$

$$X = \begin{pmatrix} V^* \\ \Omega^* \end{pmatrix}$$

The result of d will be a 3 x 1 vector. We could then use $\|d\|$ to compare with e.

Step3. Do iteration till finding desired number of inliers or end iteration.

Step4. Use inlier set to recompute the velocity

In this case, we need to select at least 3 groups of p and $\dot{p}$ to solve the motion field equation

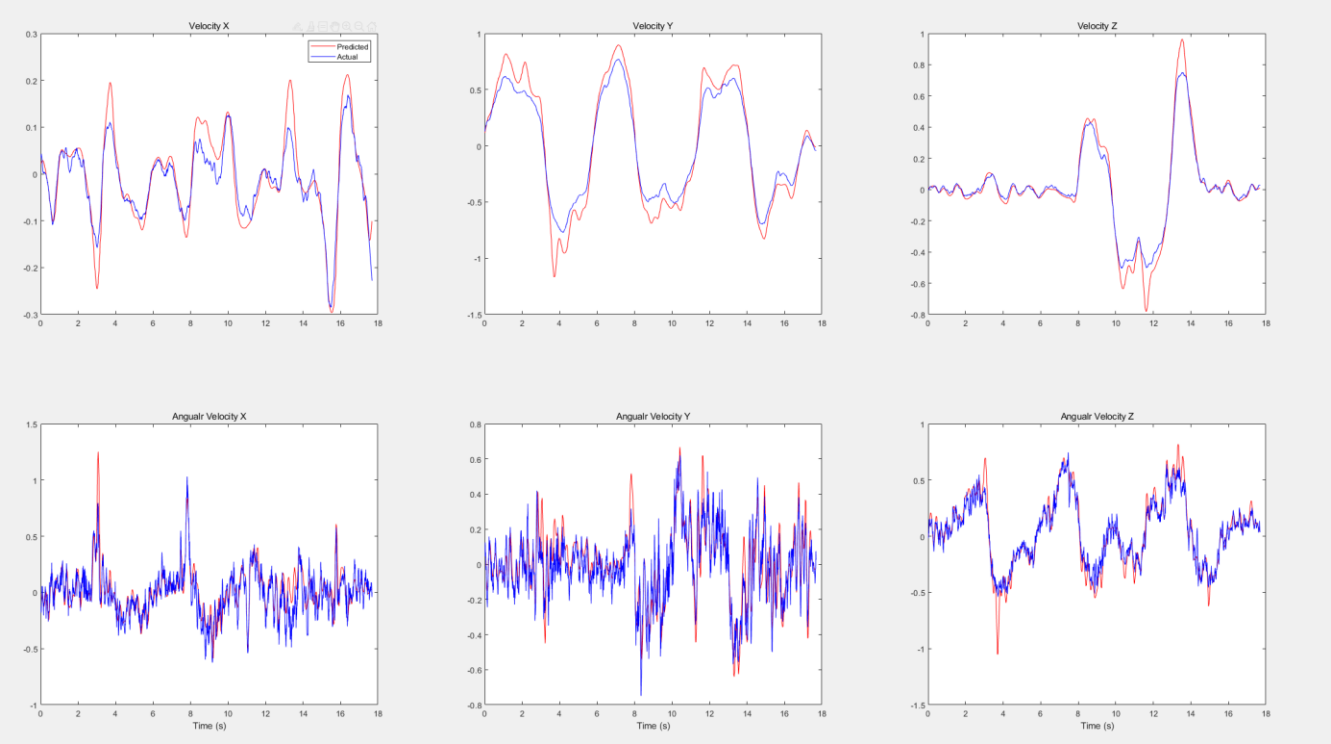
Step5. Change the frame of the computed velocity to world frame

$$Adjoint = \begin{bmatrix} R_c^w & 0 \\ 0 & R_c^w \end{bmatrix}$$

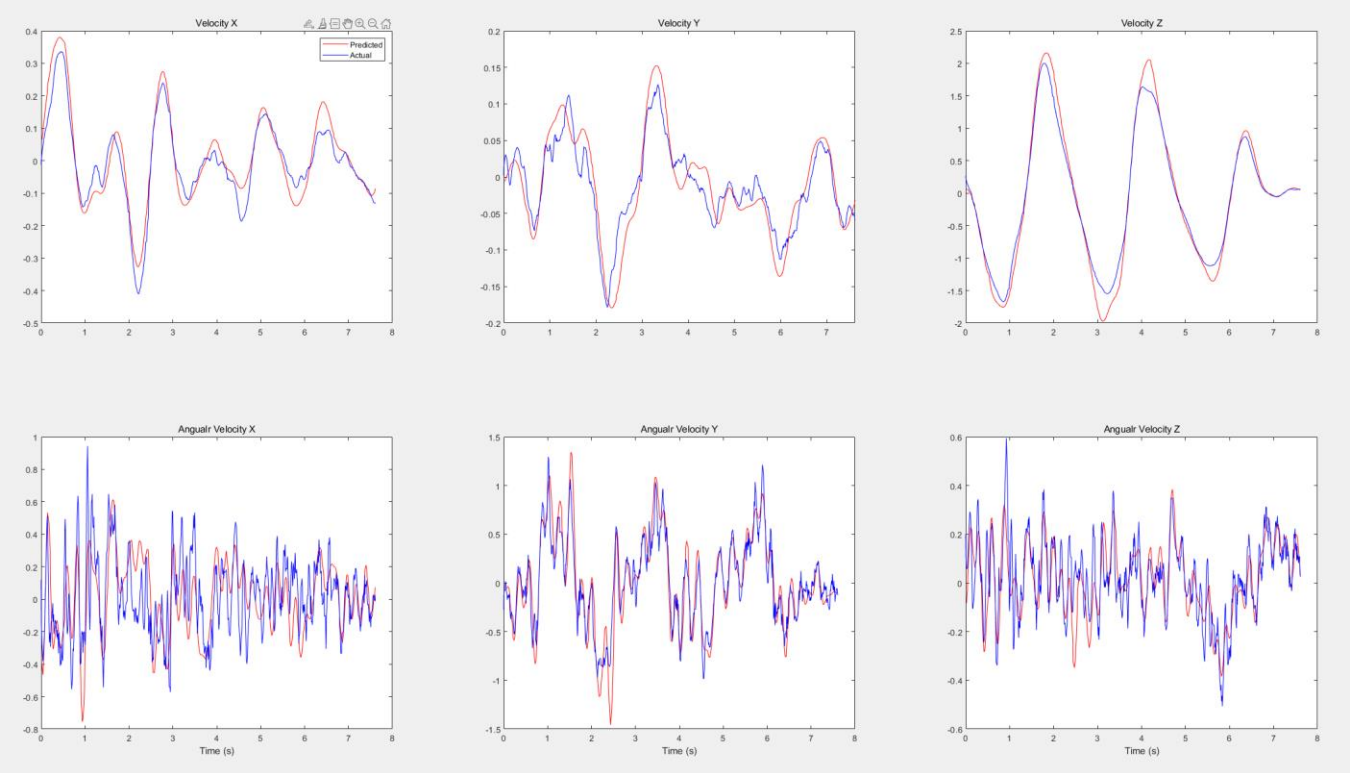
$$result = Adjoint * Velocity$$

Plotting results **with RANSAC (lowpass filter with wpass = 0.01)**:

Datasheet 1:



Datasheet 4:



Conclusion

From the results, the model optimized by the RANSAC algorithm is better than the original results to a certain extent. The curve becomes smooth after adding a low pass filter. In terms of accuracy, the degree of coincidence of the images is high, and the calculated results can be trusted to a certain extent. For the requirements of the project, the number of iterations cannot be set too high. So it should be ensured that most of the sample points are inliers in the initialization step.