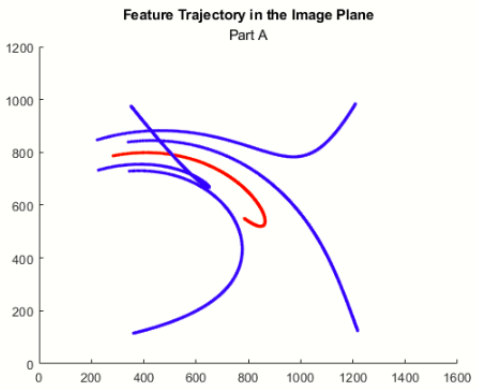
ME 599 Final Project

By Zhiyi Zhang

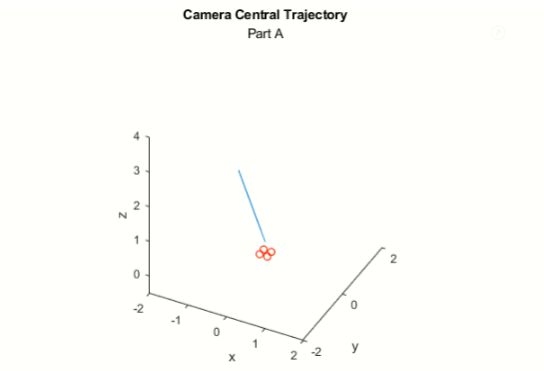
1

(a)

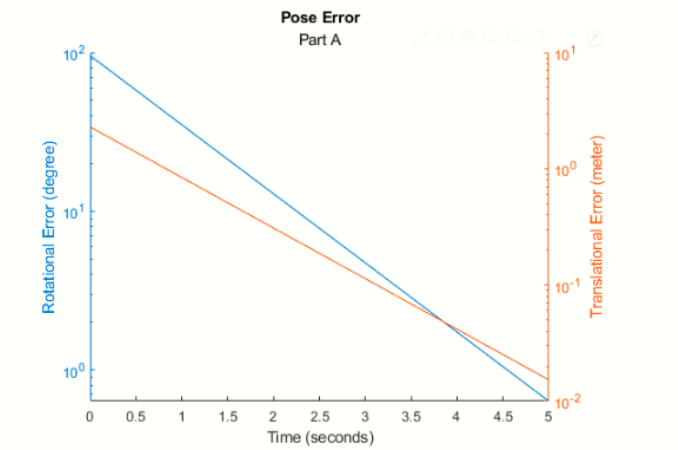
All the figures in (a) are based on PBVS algorithm. The image plane trajectory is shown as below, the red line is the trajectory of the center of the four feature points. And the four blue curves are the trajectory of the four feature points. The trajectories are curves because of the transformation of a straight line from 3D space maps into 2D image plane.



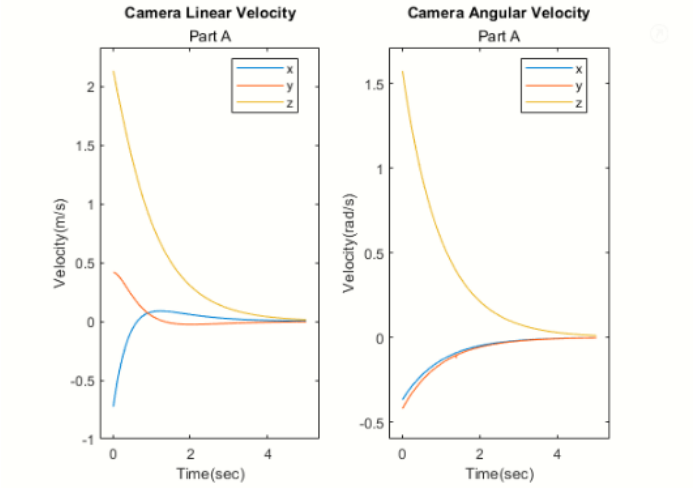
The camera central trajectory is shown as below, it is a straight line.



The pose error is shown as below, both the rotational error and the translation error are straight lines because the transform is linear.

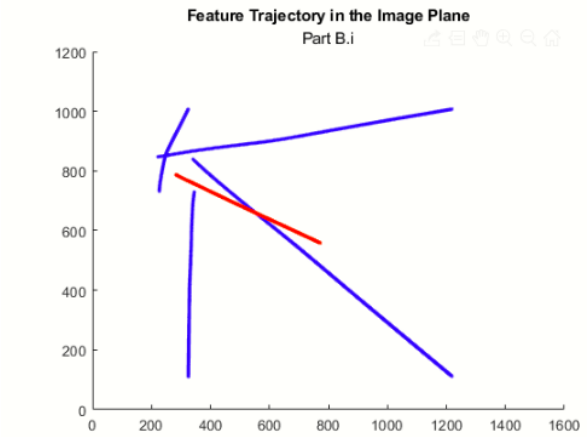


The camera velocity is shown as below. Both the linear velocity and angular velocity keep decreasing and converge to zero which is reasonable.

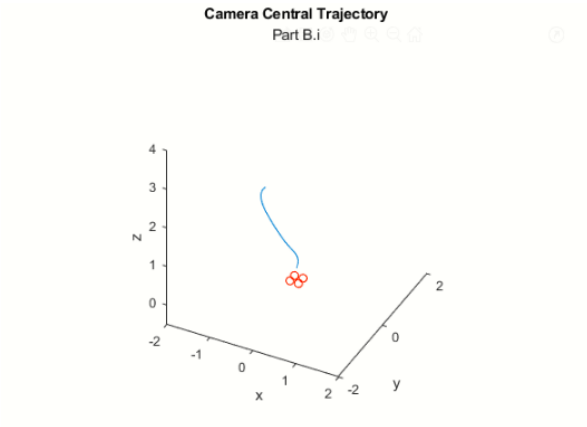


(b)

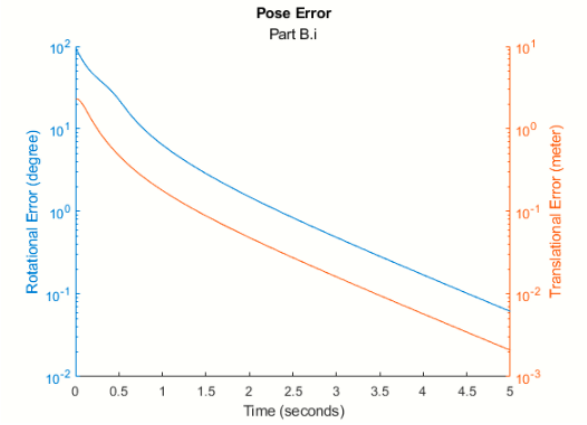
(i) All the figures in (b) are based on IBVS algorithm, the image plane trajectory is shown as below. The feature points follow the straight-line paths, and the Cartesian pose changed smoothly toward the final value.



The camera central trajectory is shown as below, it is no longer a straight line but a curve.

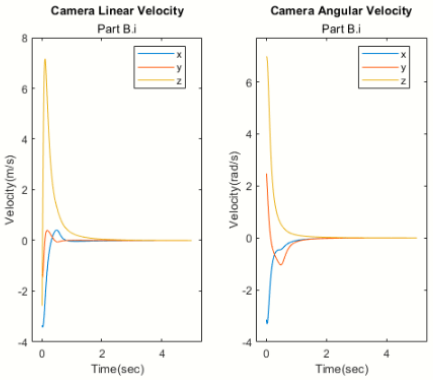


The pose error is shown as below, both the rotational error and the translation error are curves but share similar decreasing trend.

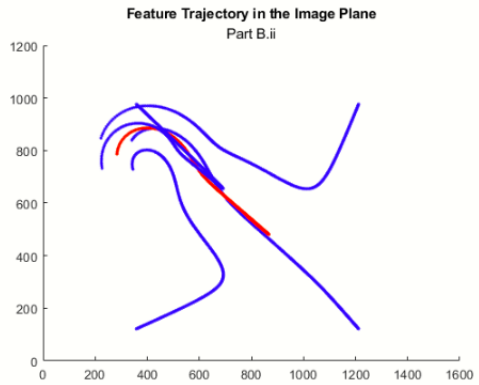


The camera velocity is shown as below. The linear velocity decrease then increase, then decrease again and converge to zero. And the angular velocity only has this trend along y axis. Its angular velocity along x and z axis keep decreasing. It makes sense when we

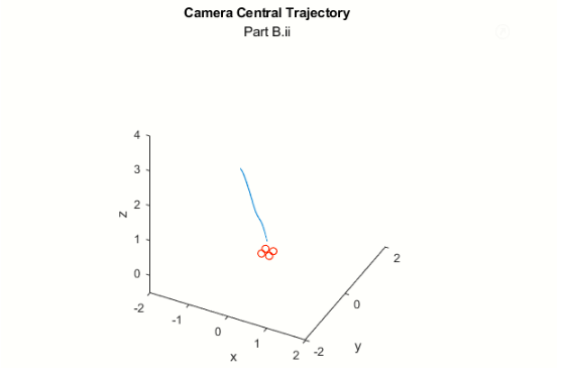
look at the camera trajectory, this change comes from two the nonlinear trajectory along y axis.



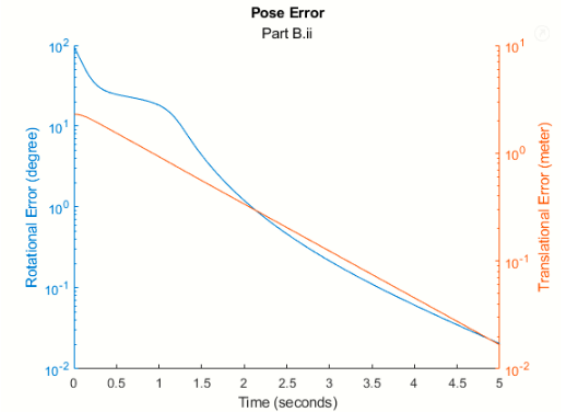
(ii)In this part, we assume the fixed Z value (true feature depth) to be 0.3 according to the relative pose of the camera with respect to the board. This means the camera motion is approximately in a plane parallel to the plane of feature points. As said in textbook, it is easy in simulation but not in reality. The image plane trajectory is shown as below. The image-plane paths are no longer straight because the Jacobian is a poor approximation of the relationship between the camera motion and image feature motion now.



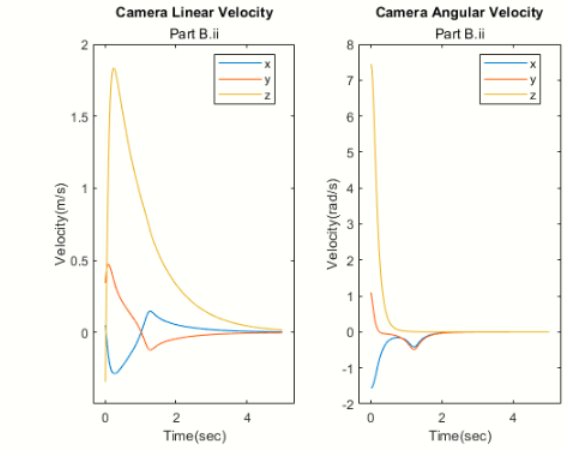
The camera central trajectory is shown as below, it is no longer a straight line but a curve.



The pose error is shown as below, both the rotational error is straight line, but the translation error is unstable initially.



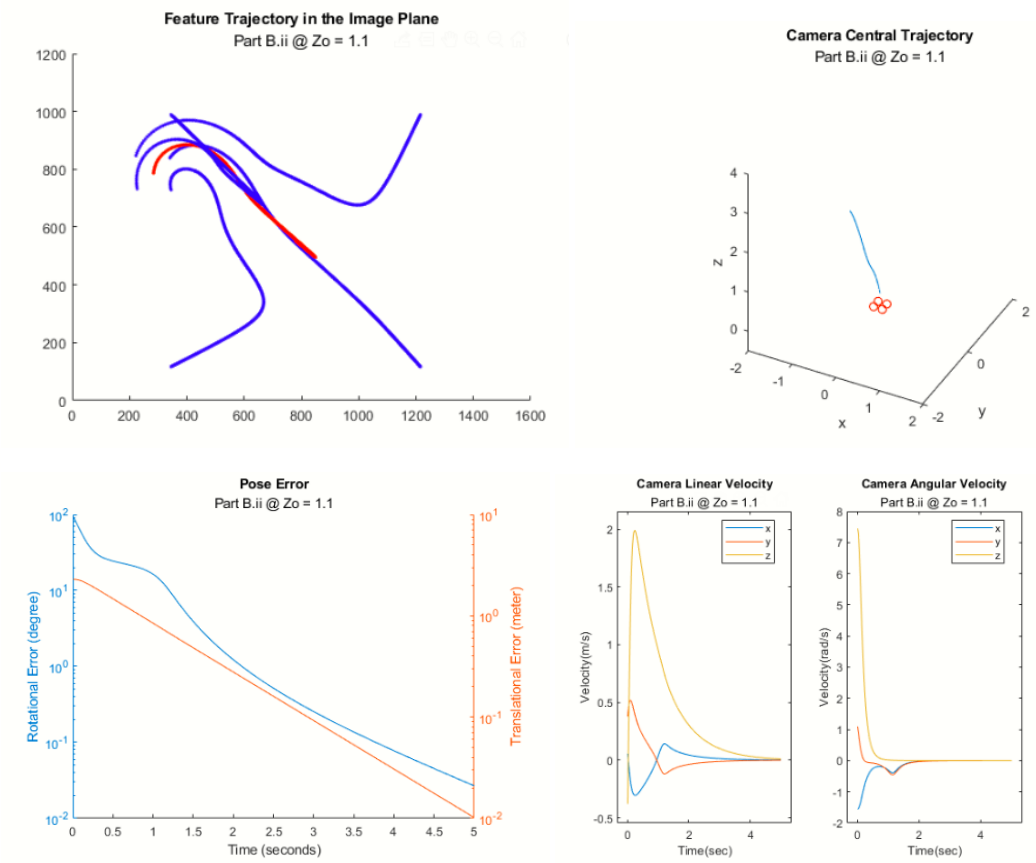
The camera velocity is shown as below. It shows the feature move in the wrong direction initially because the error in depth led to a poorly approximated Jacobian, and resulted in poor feature movement approaximation.



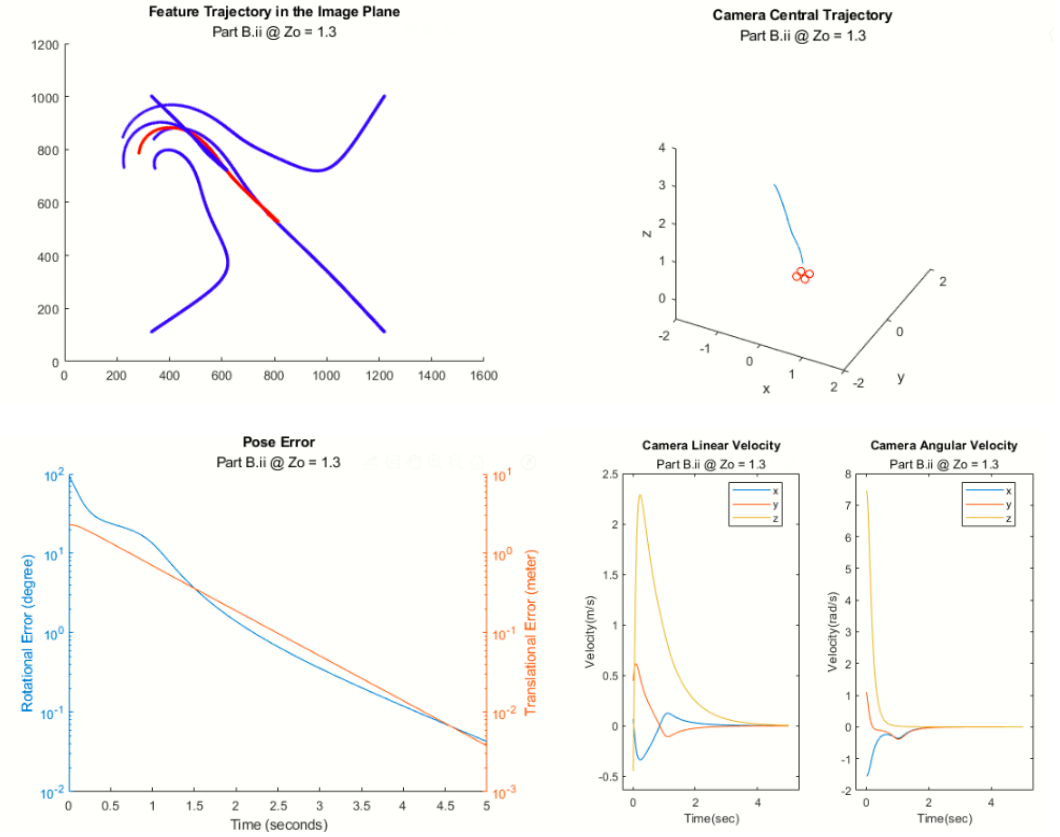
(iii)

When changed the fixed Z values to 1.1Z, 1.3Z, 1.5Z and 2Z. The image plane trajectory, camera central trajectory, pose error and camera velocity figures are shown as below.. For the pose error, the transition error looks good, but the rotational error is always greater than what in (i). The overall shape is similar to the initial Z condition. If we decrease the implemented Z values in IBVS, the more accuracy the IBVS algorithm will be. We can see when Z is small, the IBVS still converge. When Z is big, the camera displacement at each timestep is large leading to a very jagged path.

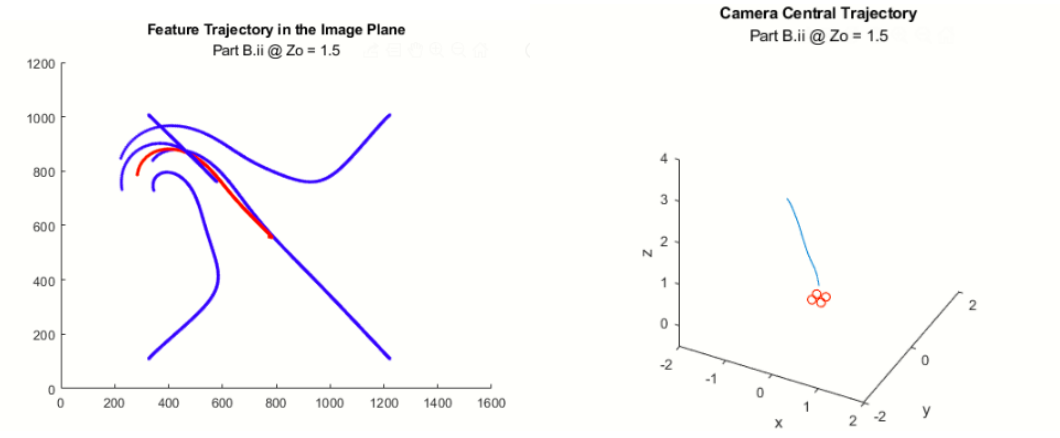
(Generated with 1.1Z)

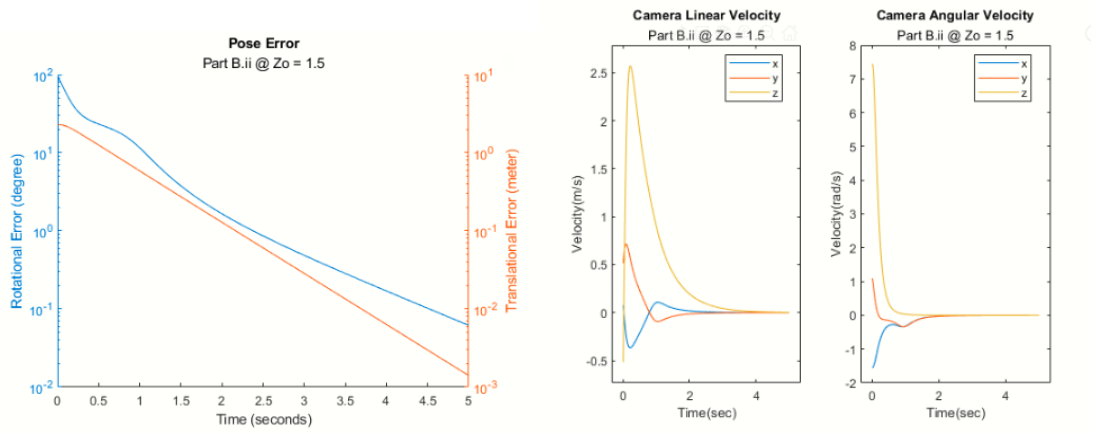


(Generated with 1.3Z)

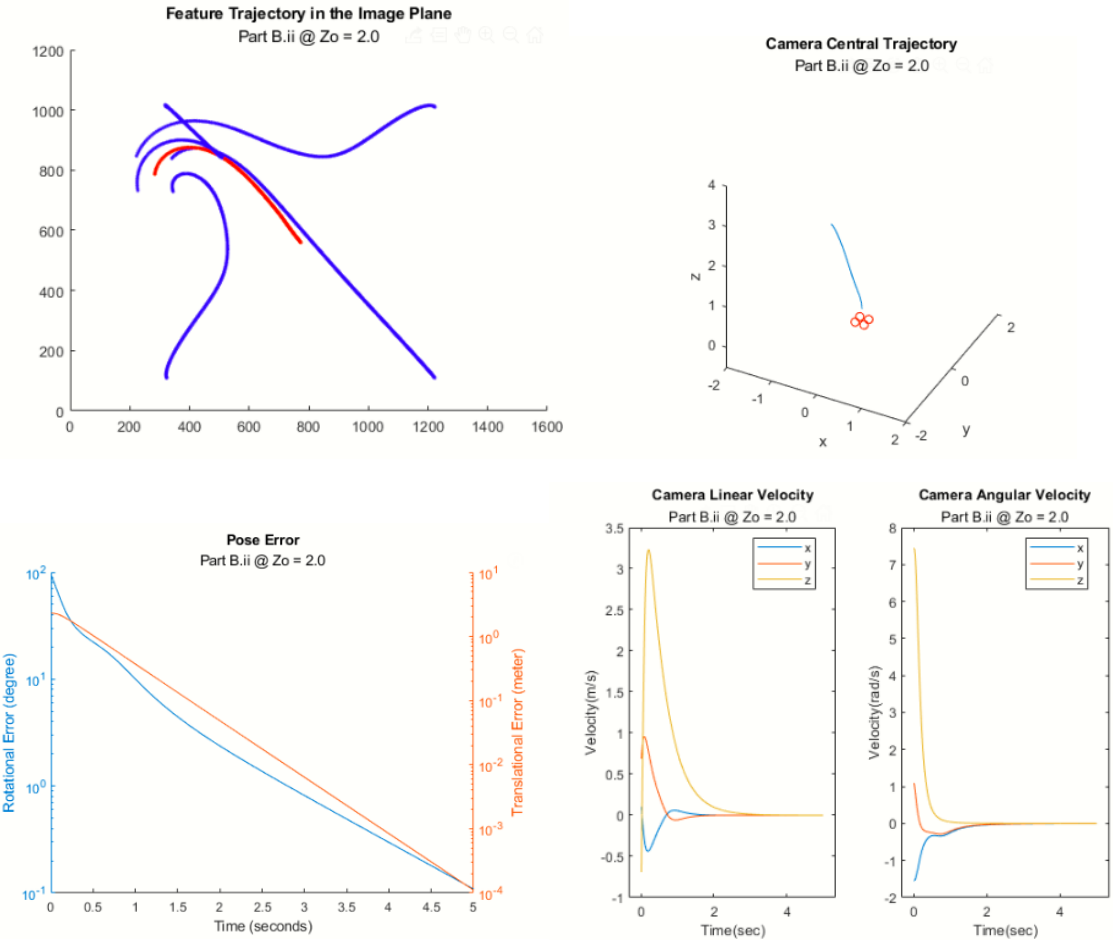


(Generated with 1.5Z)



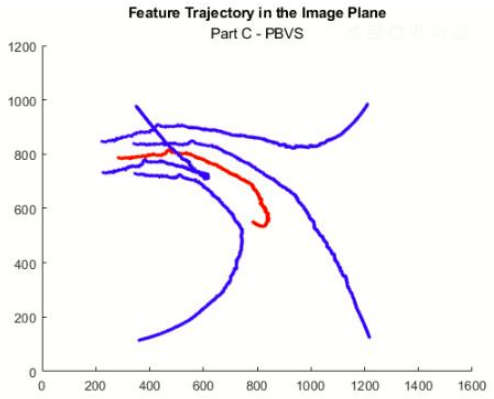


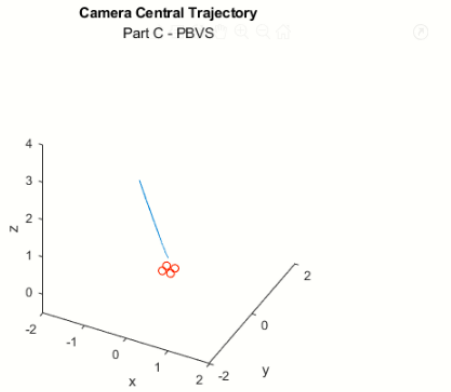
(Generated with 2Z)

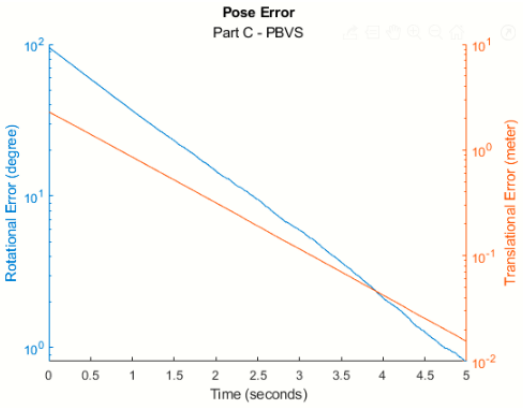


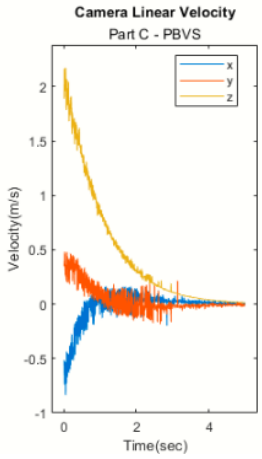
(c)

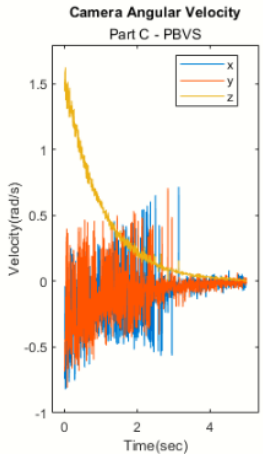
When considering a Gaussian noise, the figure with PBVS algorithm shown as below



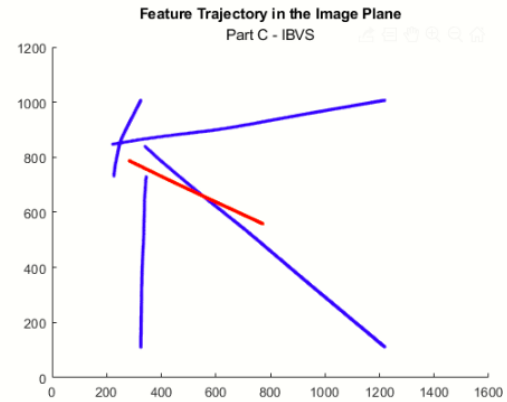


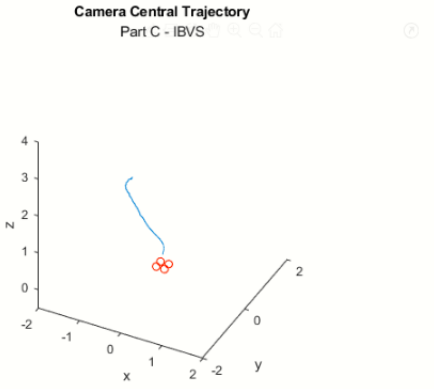


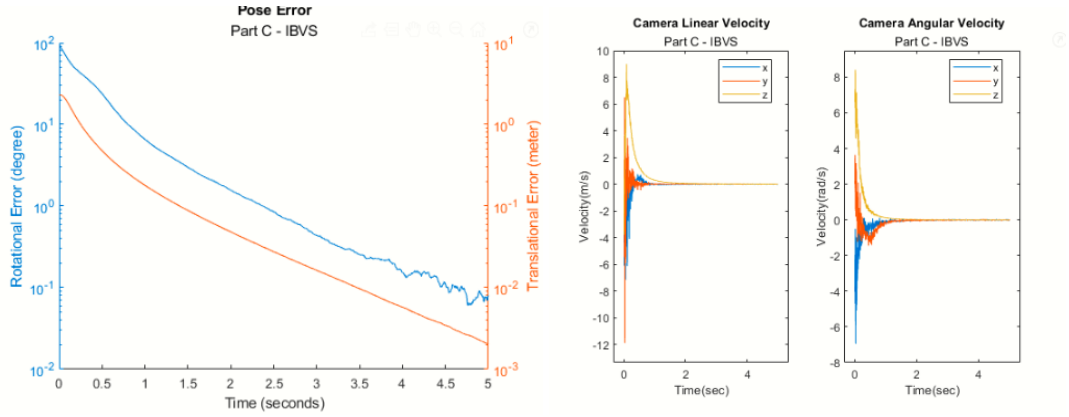




Comparing with the figures with IBVS algorithm shown as below







Compare these two sets of figure, we can find that IBVS algorithm work better when the feeature point measurements are noisy. The camera velocity doesn' t need to change frequently if use the IBVS algorithm, which makes more sense when it is done in reality.