

Figure 1.Simulink for ekf

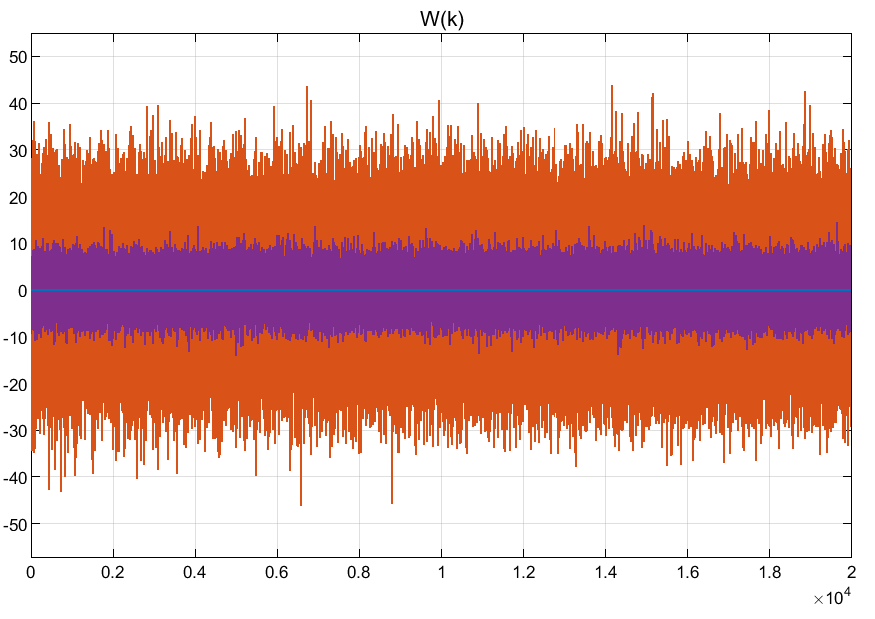


Fig 2.Process noise

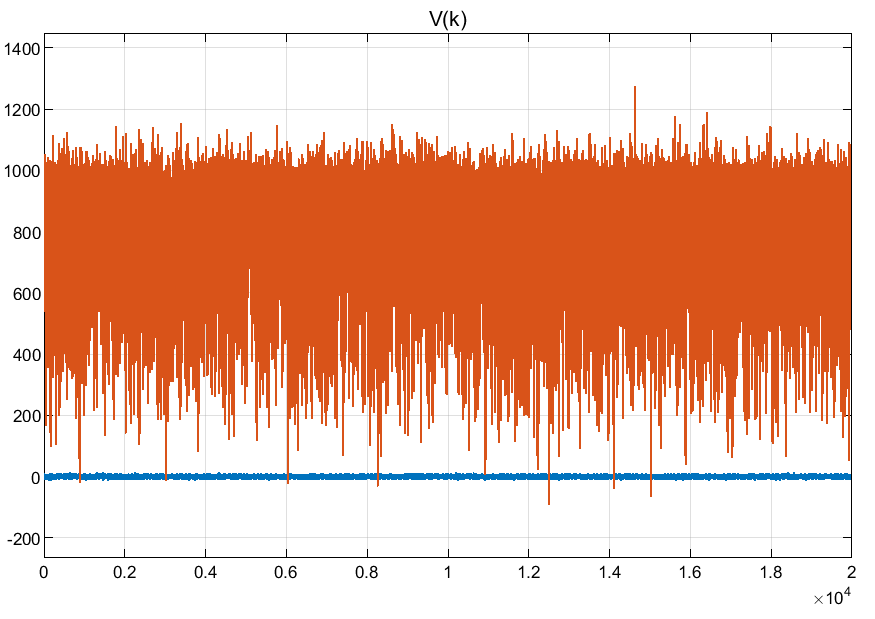


Fig3. Measurement noise

%initialize params

simlen = 40; %simulation length

stderr\_pre = zeros(8,simlen,1000); %state errors for pre state estimations

stderr\_post = zeros(8,simlen,1000); %state errors for post state estimations

std\_pre = zeros(8,simlen); %standard deviation for state errors for pre state estimations

std\_post = zeros(8,simlen); %standard deviation for state errors for post state estimations

mean\_pre = zeros(8,simlen); %mean for state errors for pre state estimations

mean\_post = zeros(8,simlen); %mean for state errors for post state estimations

for i=1:8

for j=2:simlen

for k =1:1000

stderr\_pre(i,j,k)=sterr\_pre(j+simlen\*(k-1),i); %get stderr\_pre from simulation result.

stderr\_post(i,j,k)=sterr\_post(j+simlen\*(k-1),i); %get stderr\_post from simulation result.

end

std\_pre(i,j)=std(stderr\_pre(i,j,:)); %get std for pre

std\_post(i,j)=std(stderr\_post(i,j,:)); %get std for post

mean\_pre(i,j)=mean(stderr\_pre(i,j,:)); %get mean for pre

mean\_post(i,j)=mean(stderr\_post(i,j,:)); %get mean for post

end

end

%figures for standard deviation and mean values at each time step for pre and post state estimations.

for i=1:8

figure(i);

subplot(4,1,1);

plot(mean\_pre(i,:));

subplot(4,1,2);

plot(mean\_post(i,:));

subplot(4,1,3);

plot(std\_pre(i,:));

subplot(4,1,4);

plot(std\_post(i,:));

end

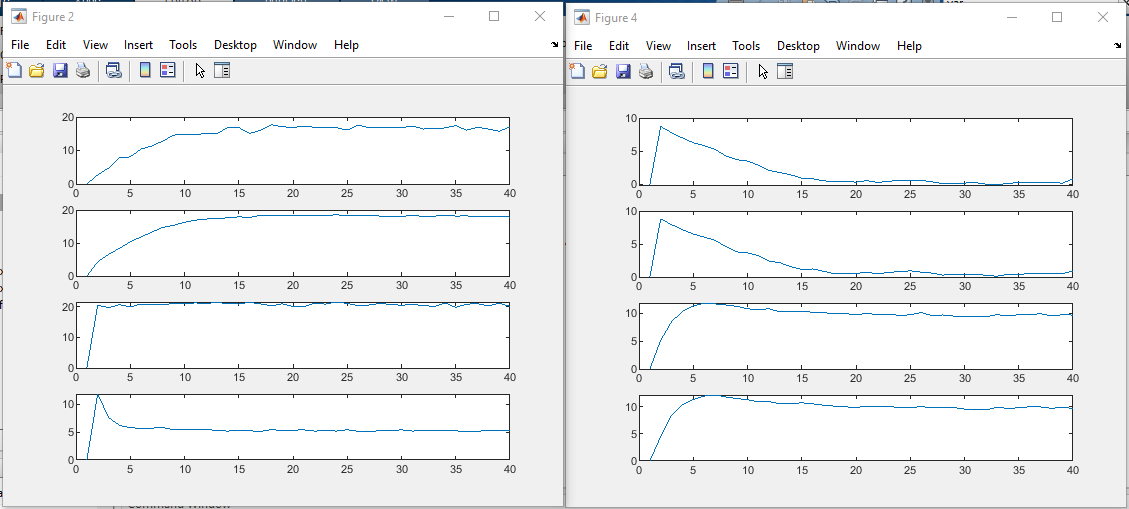


Fig 4. Mean and standard deviation for pre and post estimations.

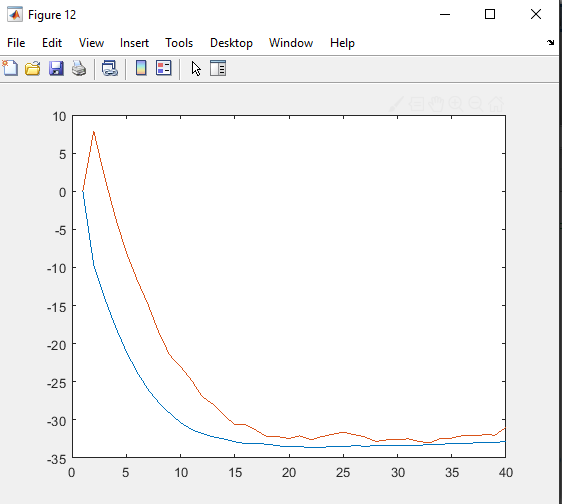


Figure 5.co-plot for standard deviation of vy

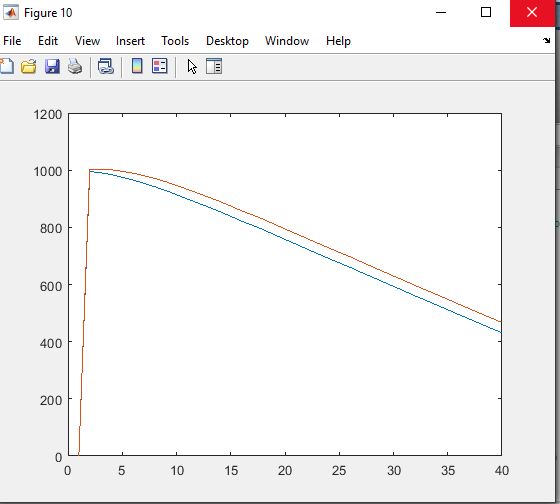


Figure 6.co-plot for standard deviation of ry