Laboratory 3 for ABE 498

Total points: 100

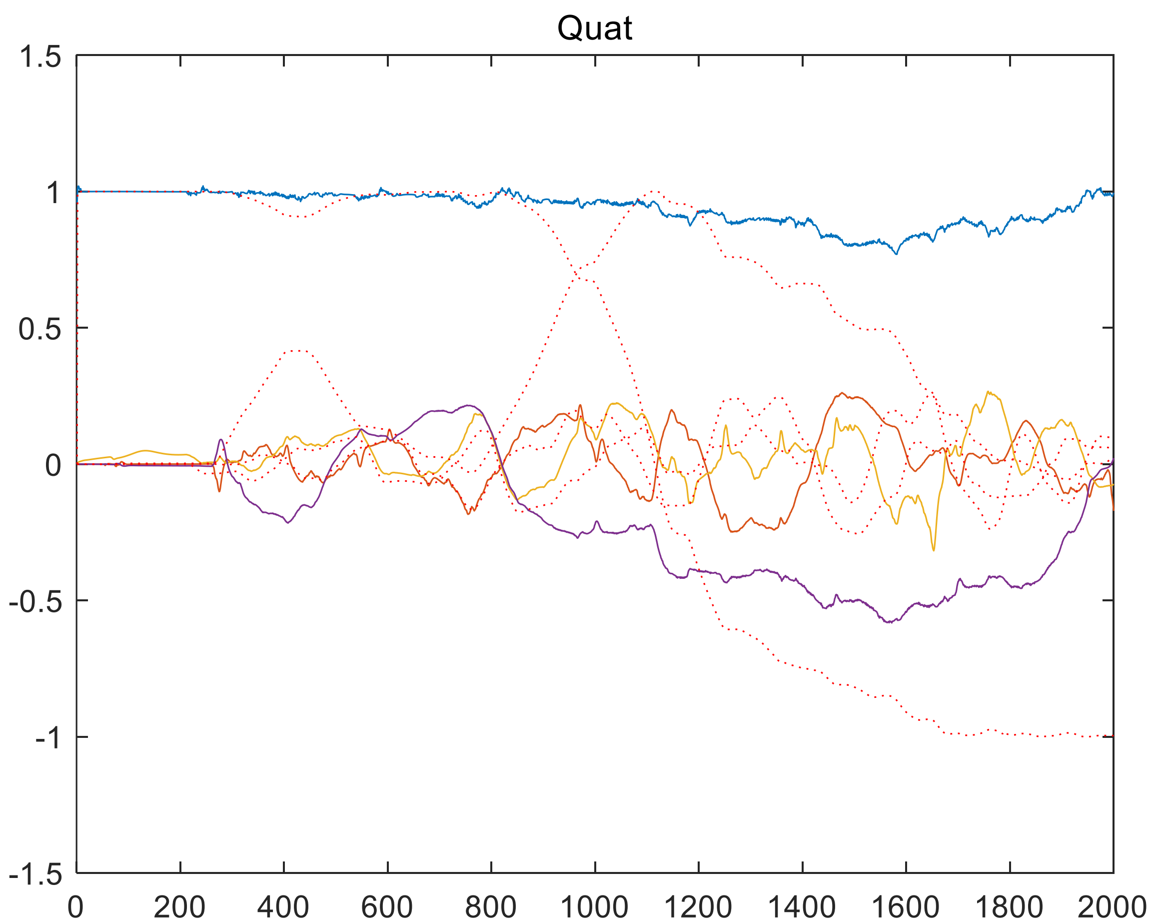
In this laboratory, we will implement a full-blown GPS-INS Extended Kalman filter

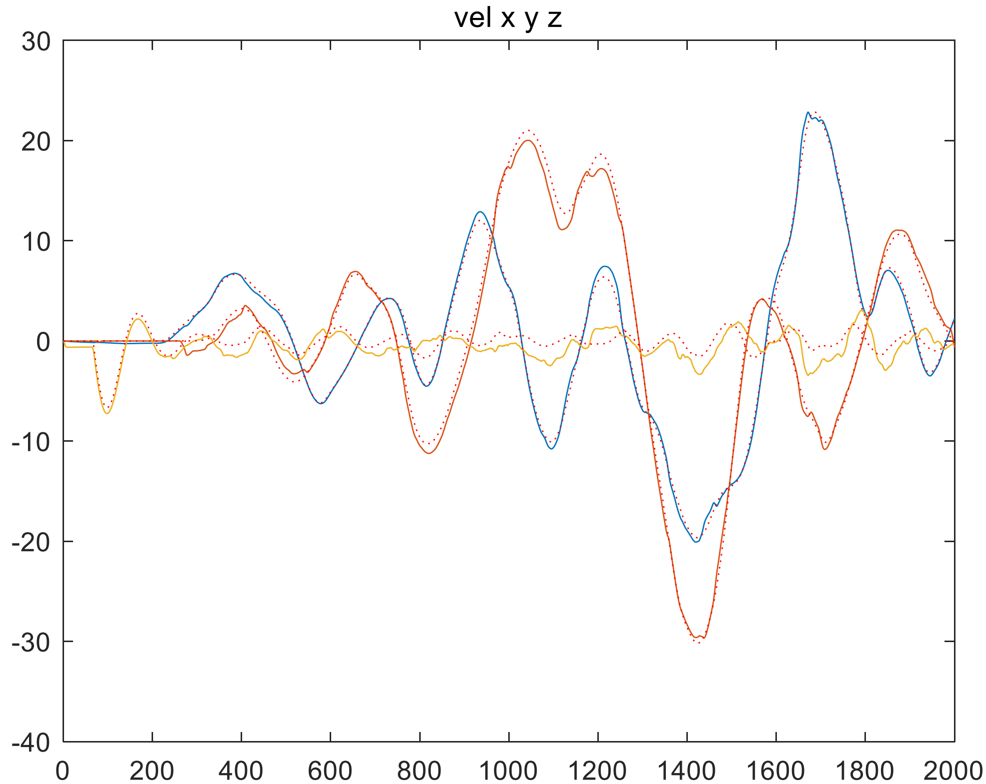
The learning outcomes are:

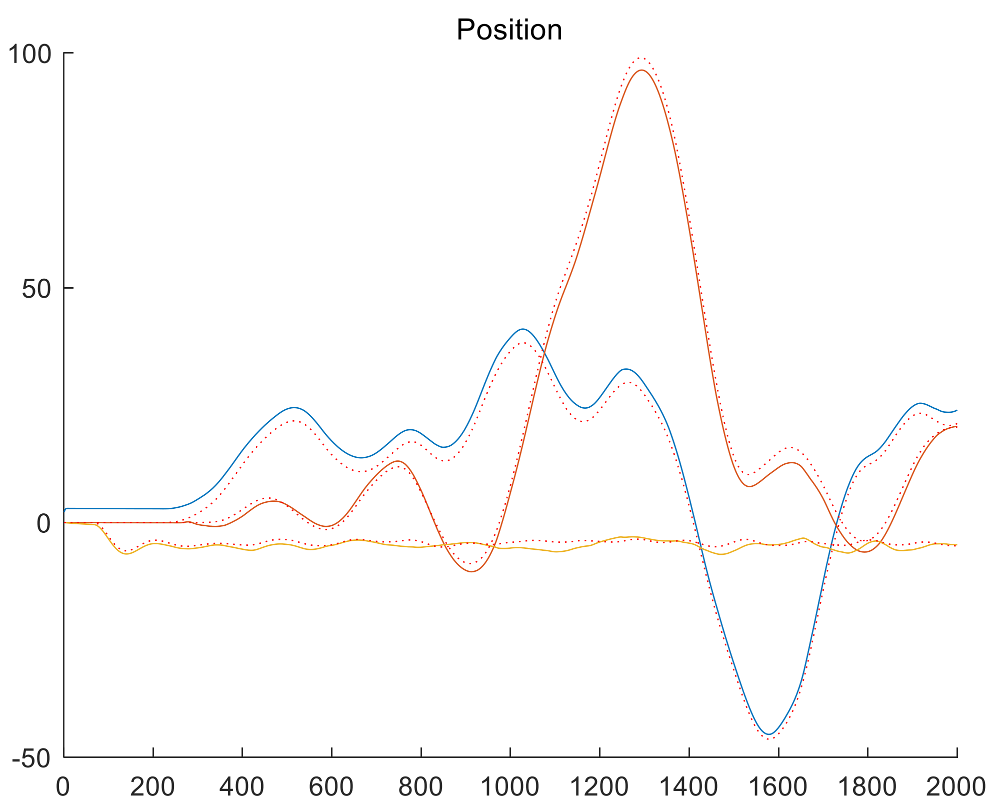
1. Understand Kalman filters and EKF
2. Implement Extended Kalman filter
3. Write software for EKF with GPS-INS integration
4. Present report and analyze results for robustness

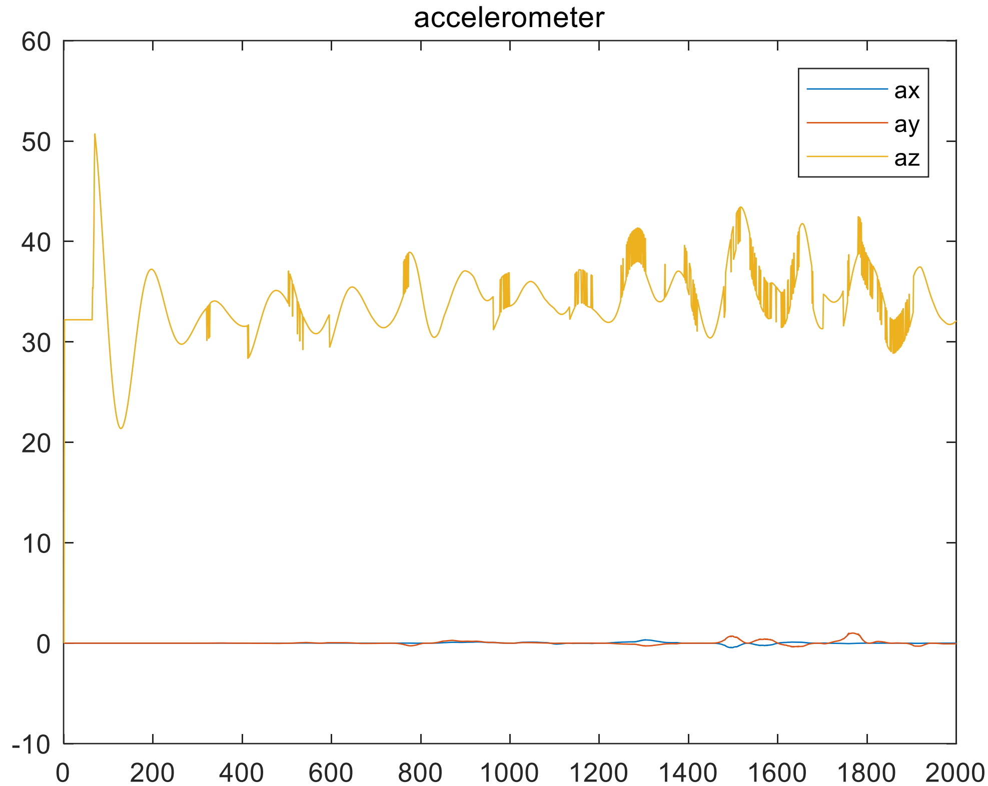
Questions for the lab report and points:

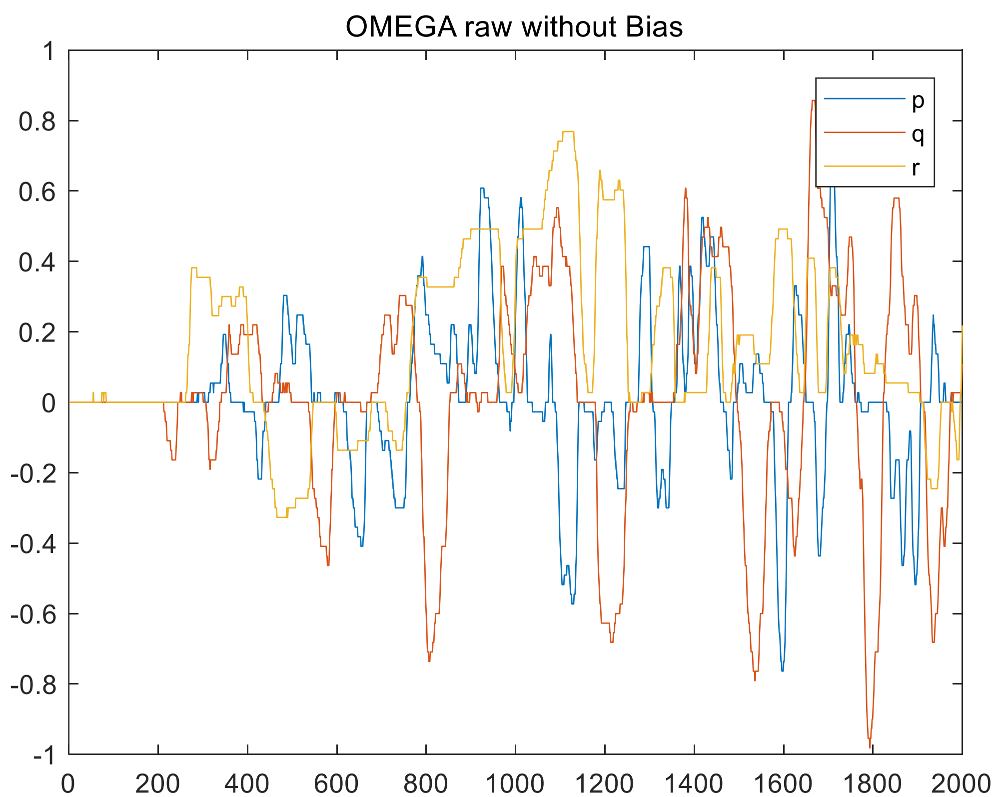
Q1: Implement the Extended Kalman Filter with the given data as instructed in the lab (80 points)

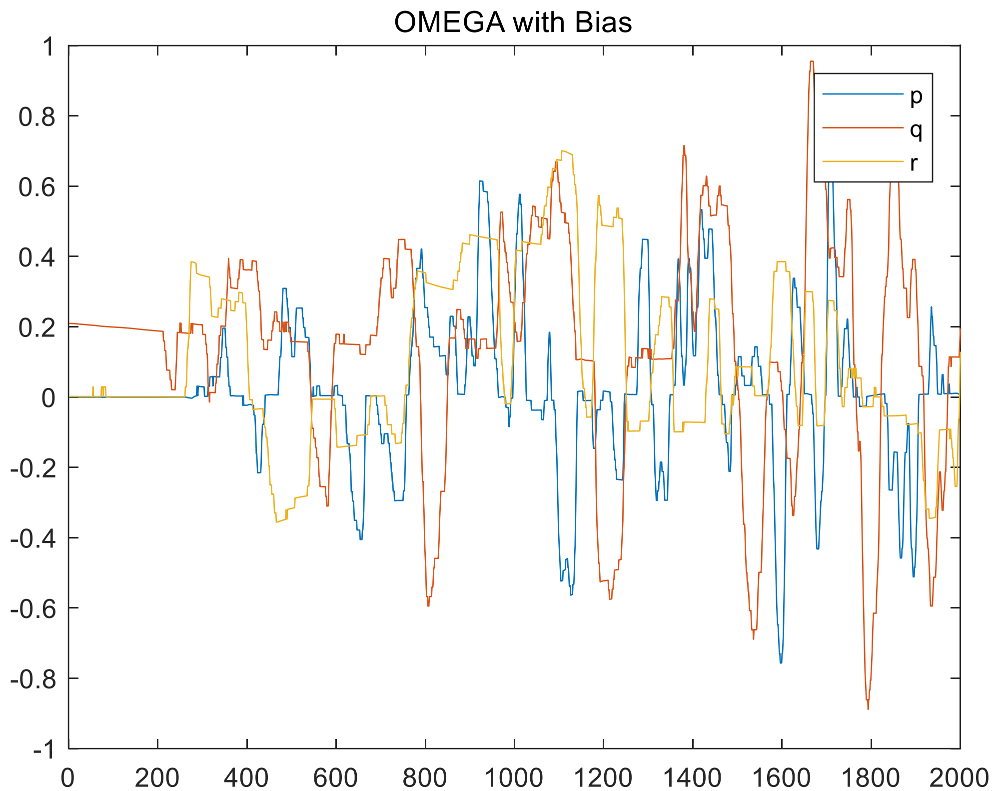


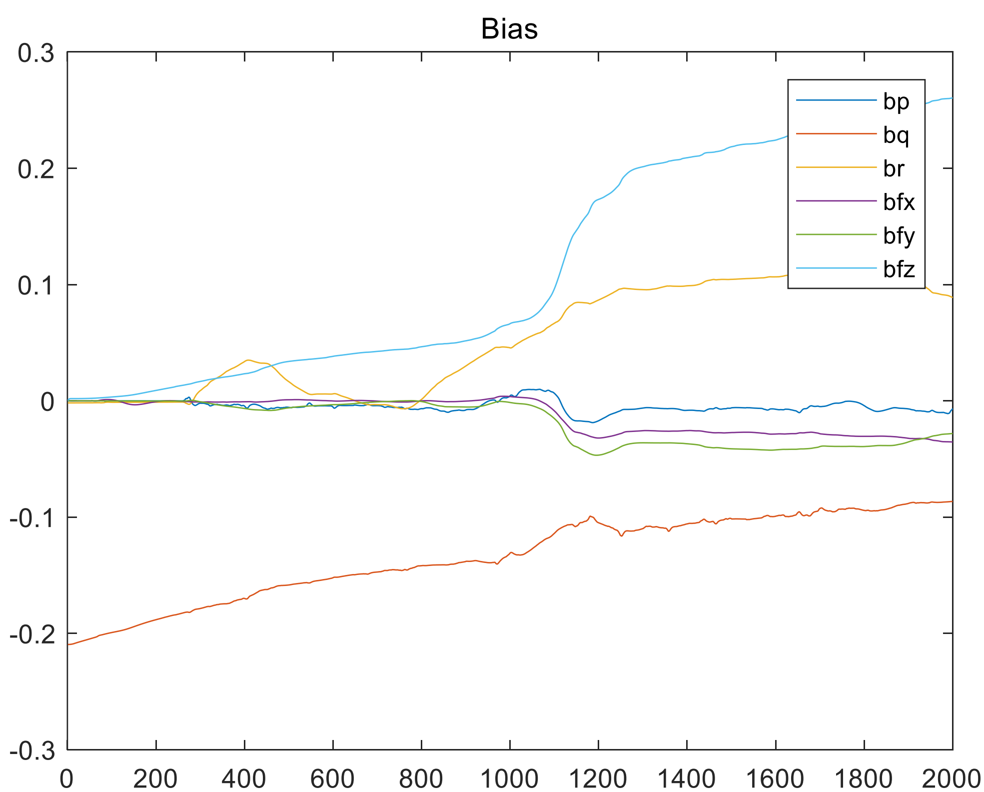

vel x y z
30
20
10
0
-10
-20
-30
-40
0
200
400
600
800
1000
1200
1400
1600
1800
2000

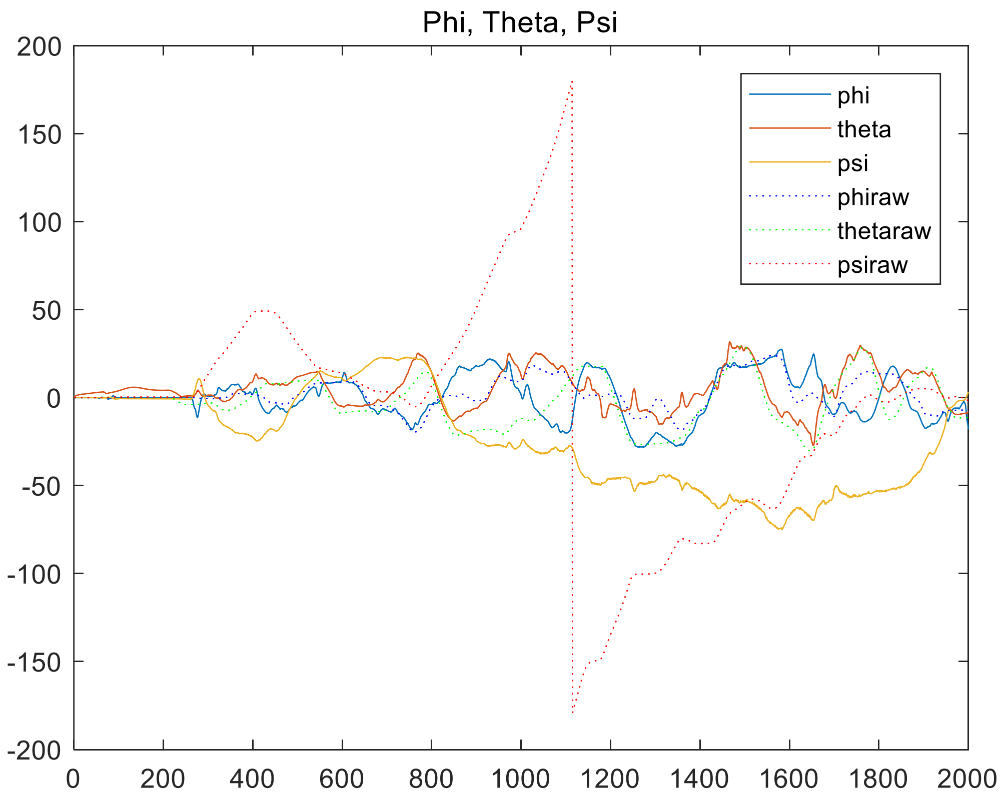

Position
100
50
0
-50
0
200
400
600
800
1000
1200
1400
1600
1800
2000


accelerometer
ax
ay
az
60
50
40
30
20
10
0
-10
0
200
400
600
800
1000
1200
1400
1600
1800
2000

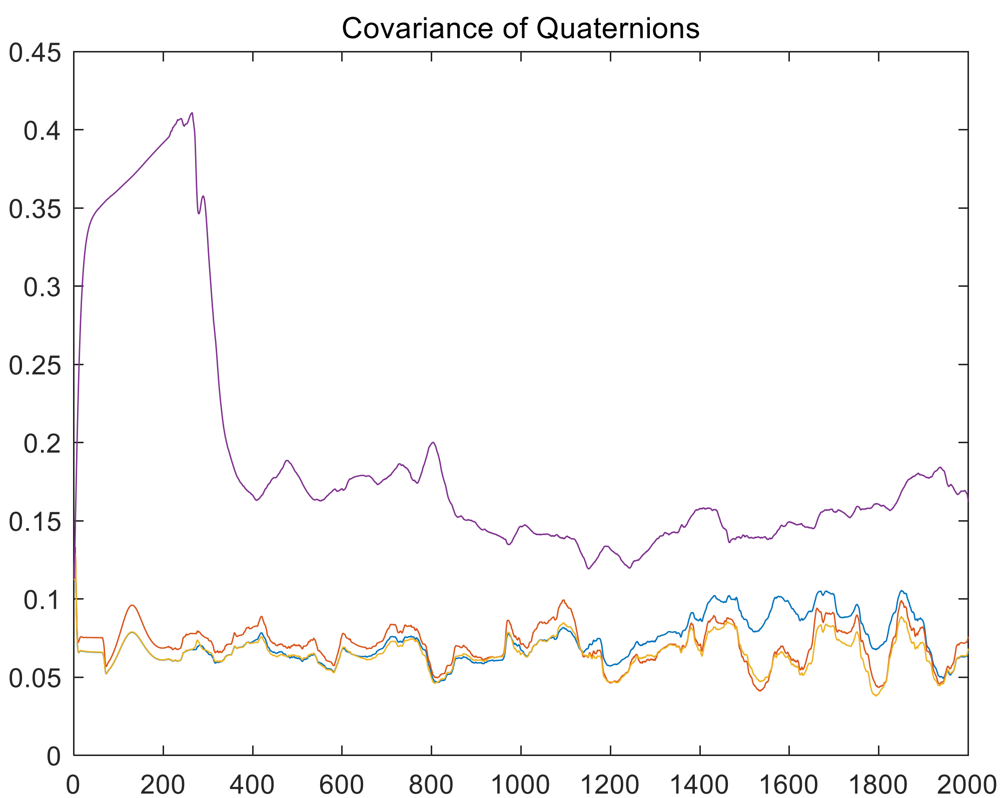

OMEGA raw without Bias
p
q
r
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
200
400
600
800
1000
1200
1400
1600
1800
2000


OMEGA with Bias
p
q
r
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
200
400
600
800
1000
1200
1400
1600
1800
2000

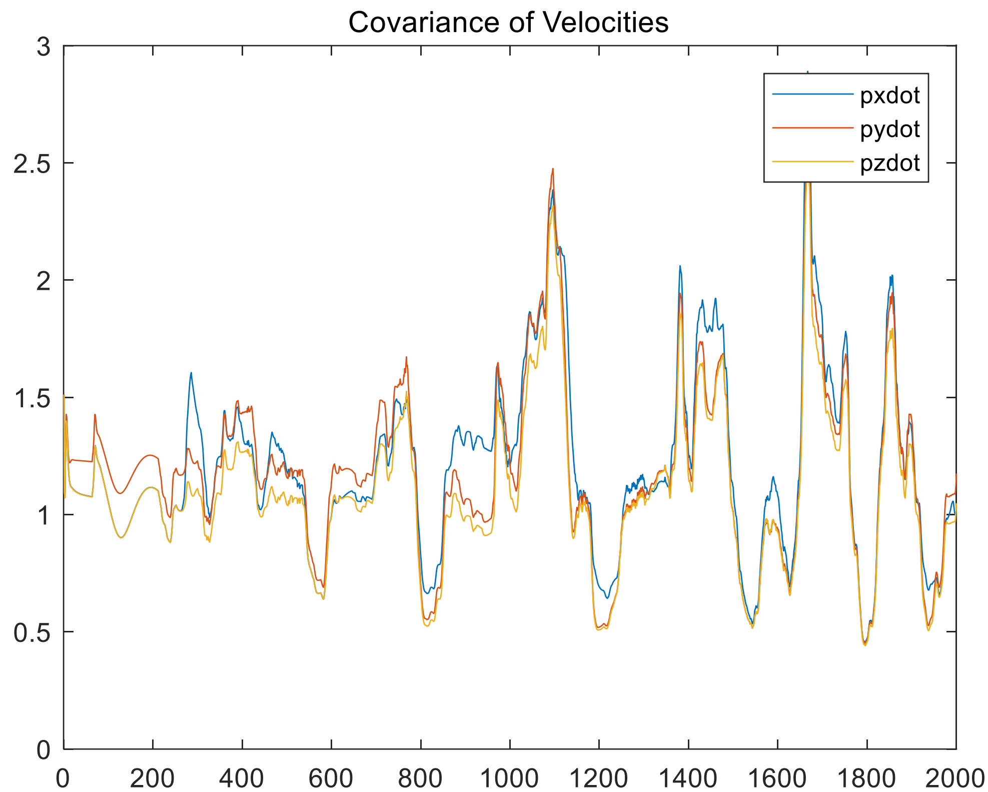

Bias
bp
bq
br
bfx
bfy
bfz
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
0
200
400
600
800
1000
1200
1400
1600
1800
2000



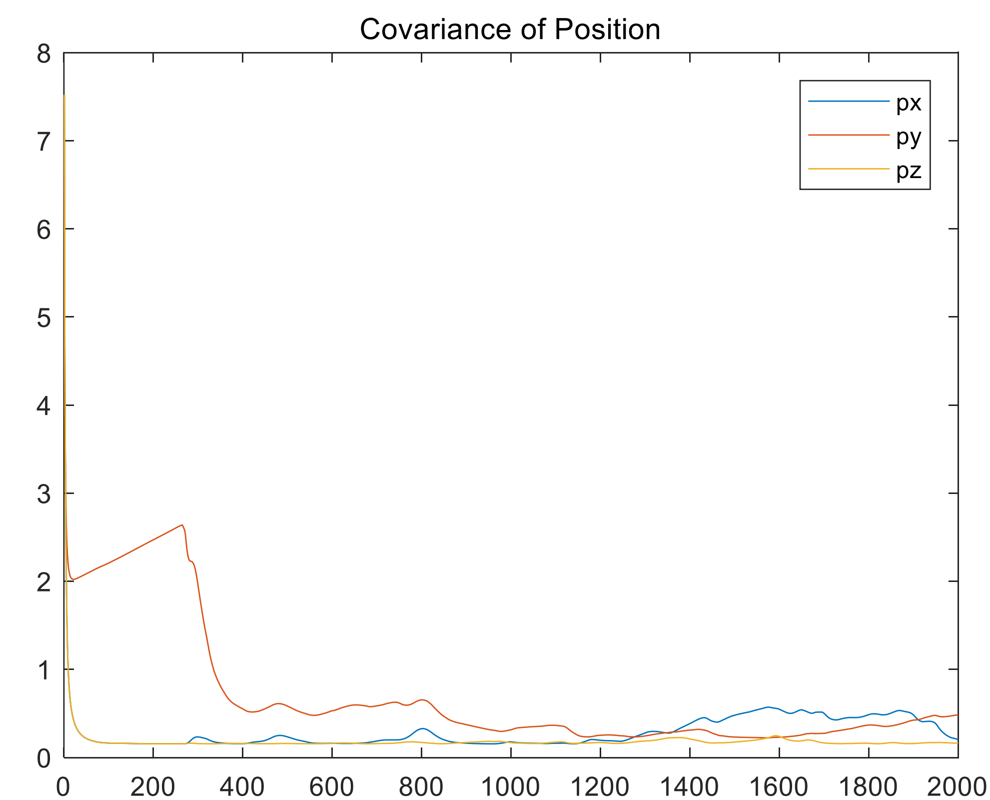
Phi, Theta, Psi
phi
theta
psi
phiraw
thetaraw
psiraw



Covariance of Quaternions



Covariance of Velocities
pxdot
pydot
pzdot
3
2.5
2
1.5
1
0.5
0
0
200
400
600
800
1000
1200
1400
1600
1800
2000

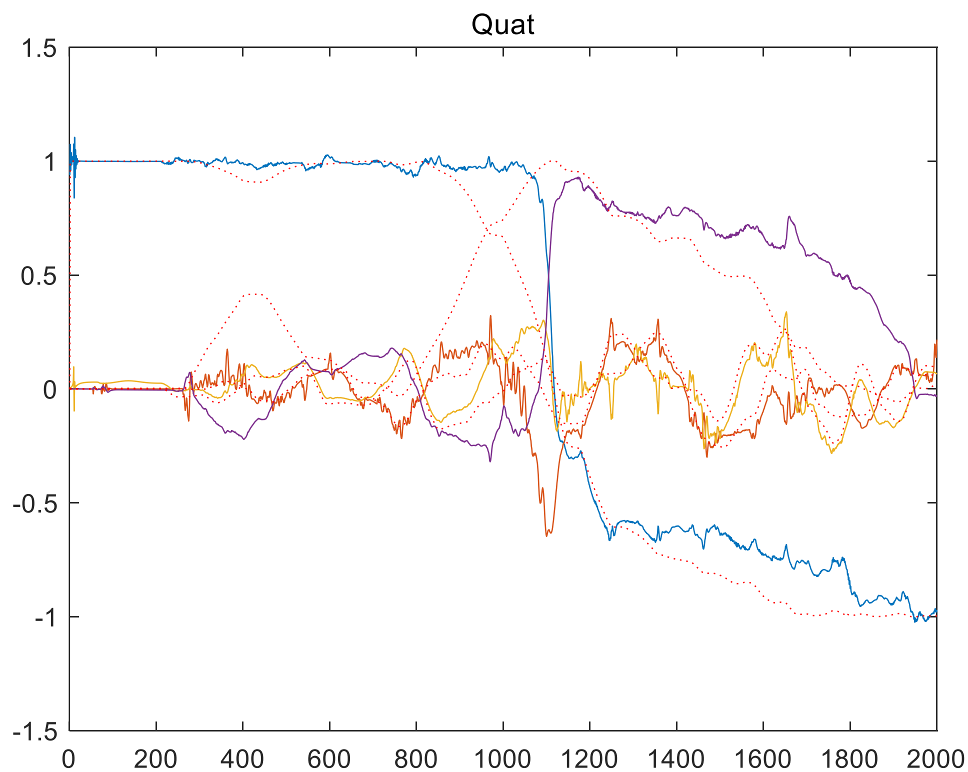


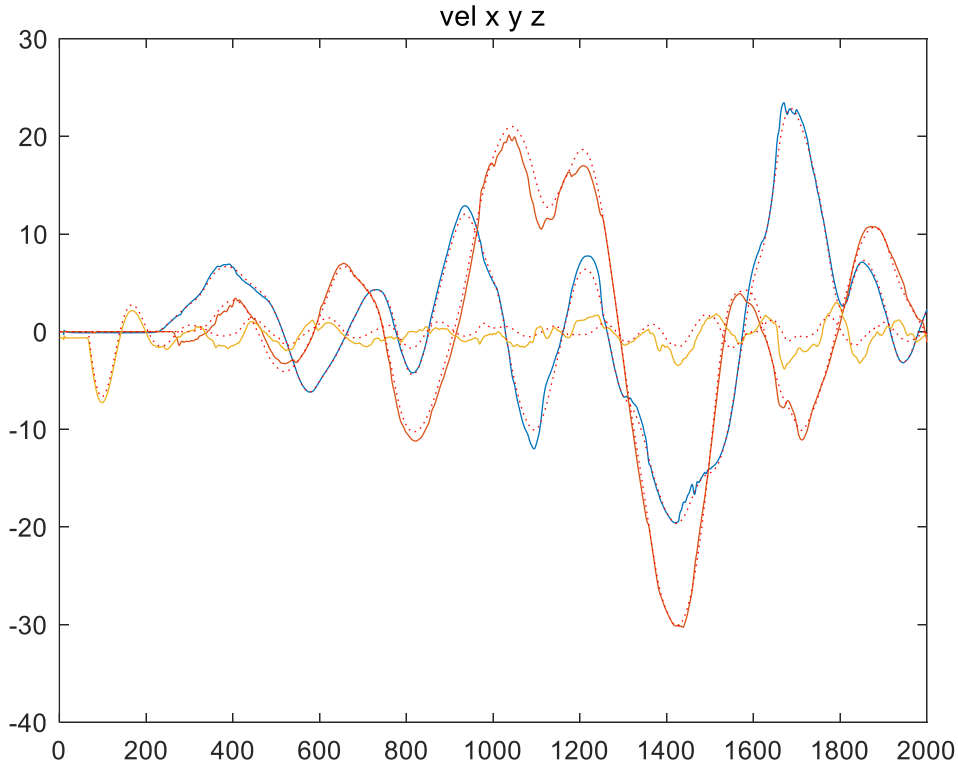
Covariance of Position
px
py
pz
8
7
6
5
4
3
2
1
0
0
200
400
600
800
1000
1200
1400
1600
1800
2000

Q2: Evaluate the performance of the Kalman filter for different values of Q and R matrices (10 points), in particular, try the following different cases and qualitatively describe what happens to the estimation errors and covariance. Is the filter fairly robust to Q and R?:

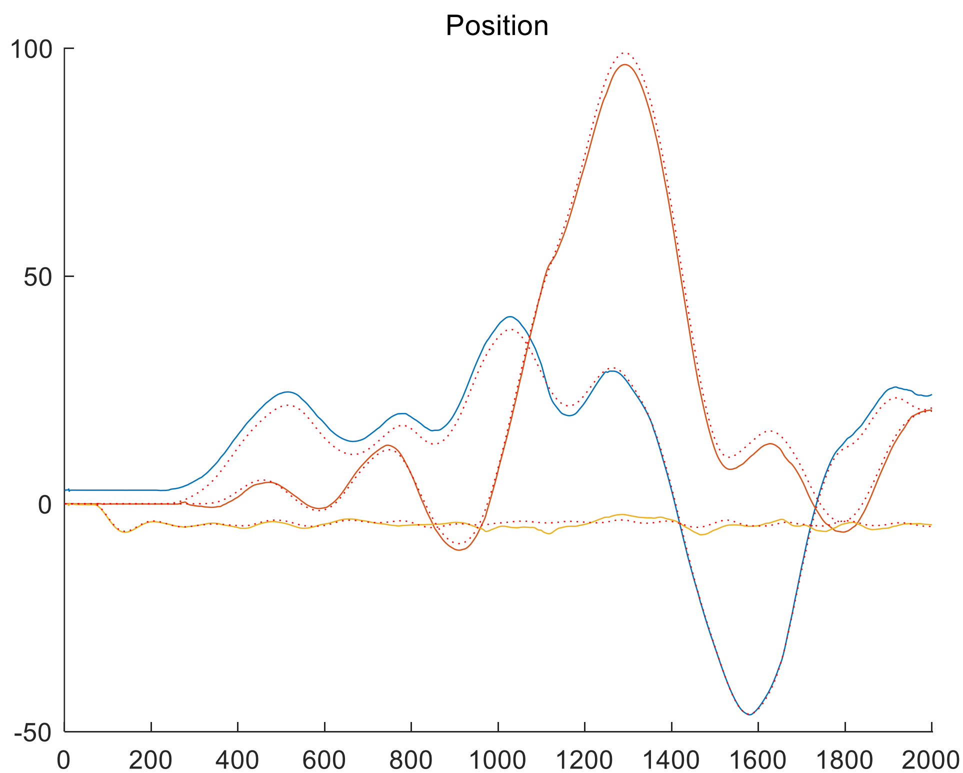
$$R = [.1\ .1\ .1\ .1\ .1\ .1]$$
$$R = [.01\ .01\ .01\ .01\ .01\ .01]$$
$$R = [10\ 10\ 10\ 8\ 8\ 8]$$
$$Q = 0.0001 * Q_{original}$$
$$Q = 100 * Q_{original}$$

1. $R = [.1\ .1\ .1\ .1\ .1\ .1]$

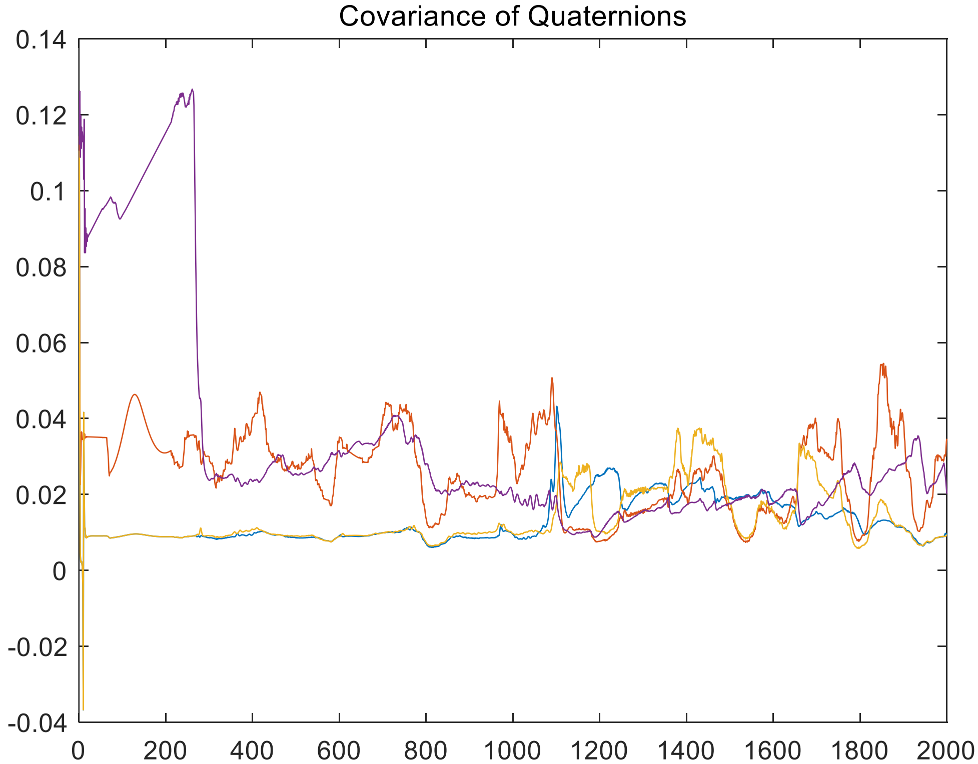


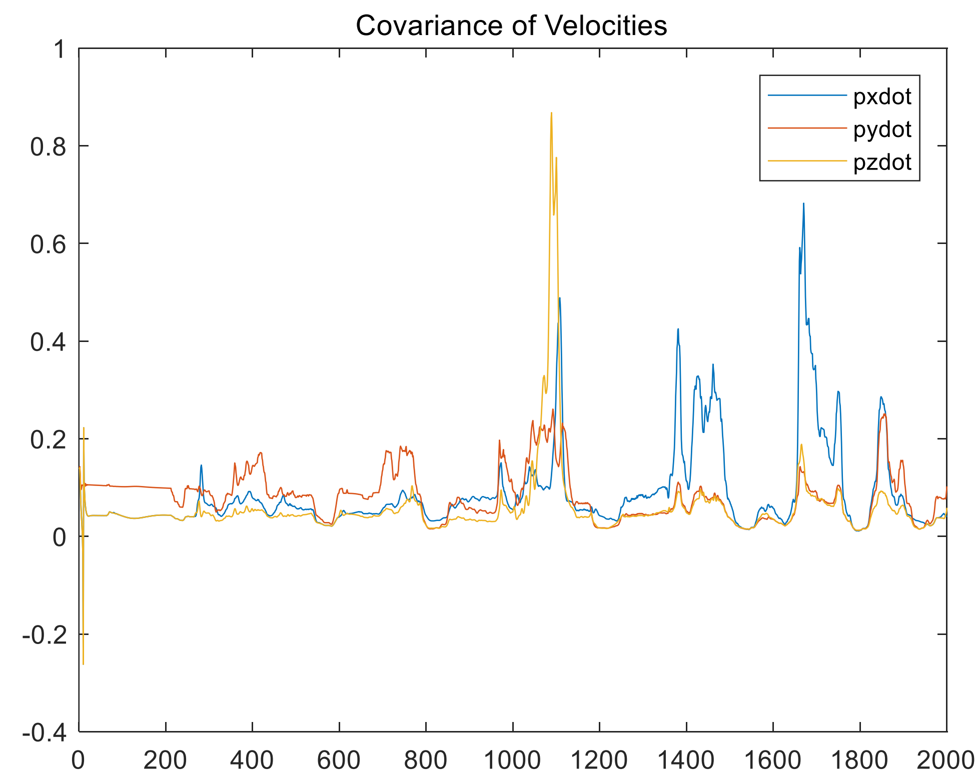


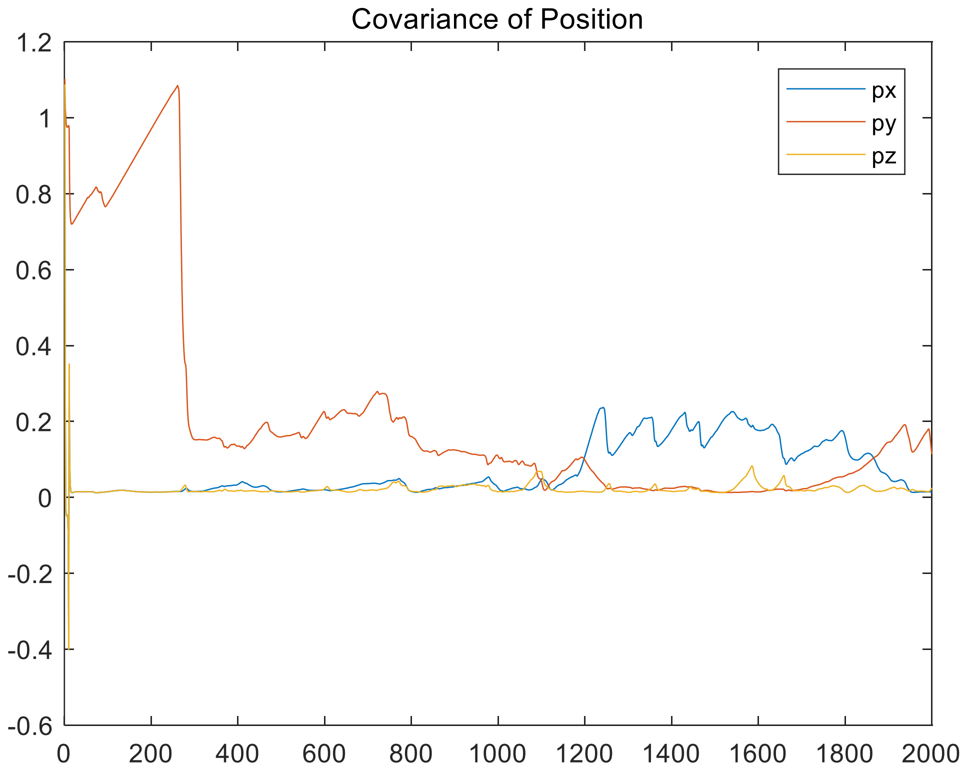
vel x y z
30
20
10
0
-10
-20
-30
-40
0
200
400
600
800
1000
1200
1400
1600
1800
2000



Position
100
50
0
-50
0
200
400
600
800
1000
1200
1400
1600
1800
2000

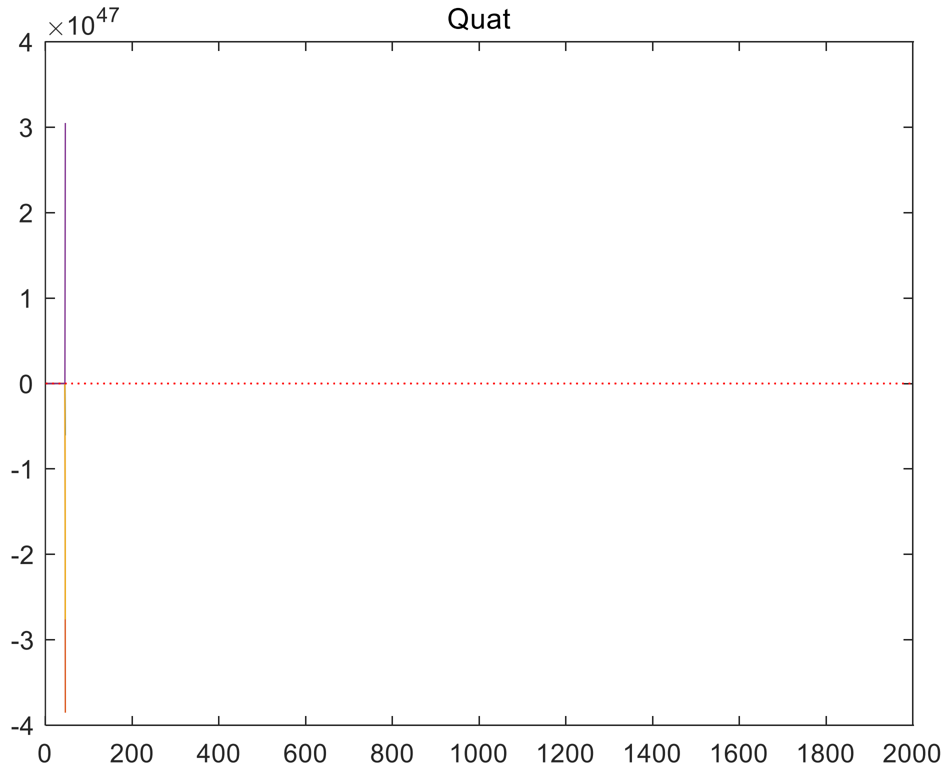

Covariance of Quaternions
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
-0.02
-0.04
0
200
400
600
800
1000
1200
1400
1600
1800
2000

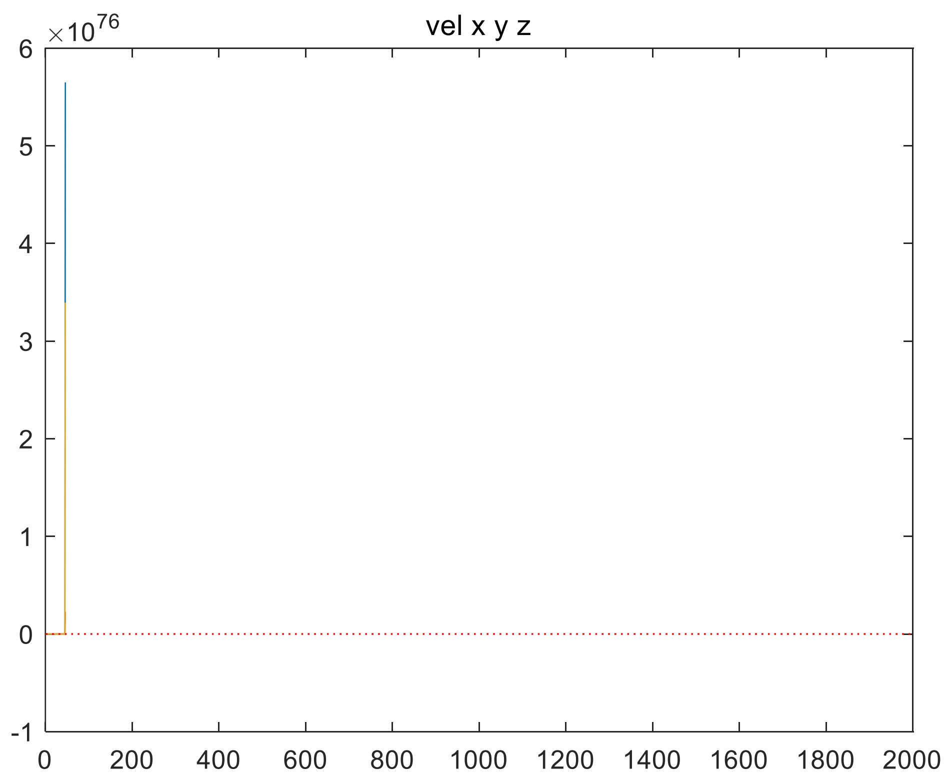

Covariance of Velocities
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
0
200
400
600
800
1000
1200
1400
1600
1800
2000
pxdot
pydot
pzdot

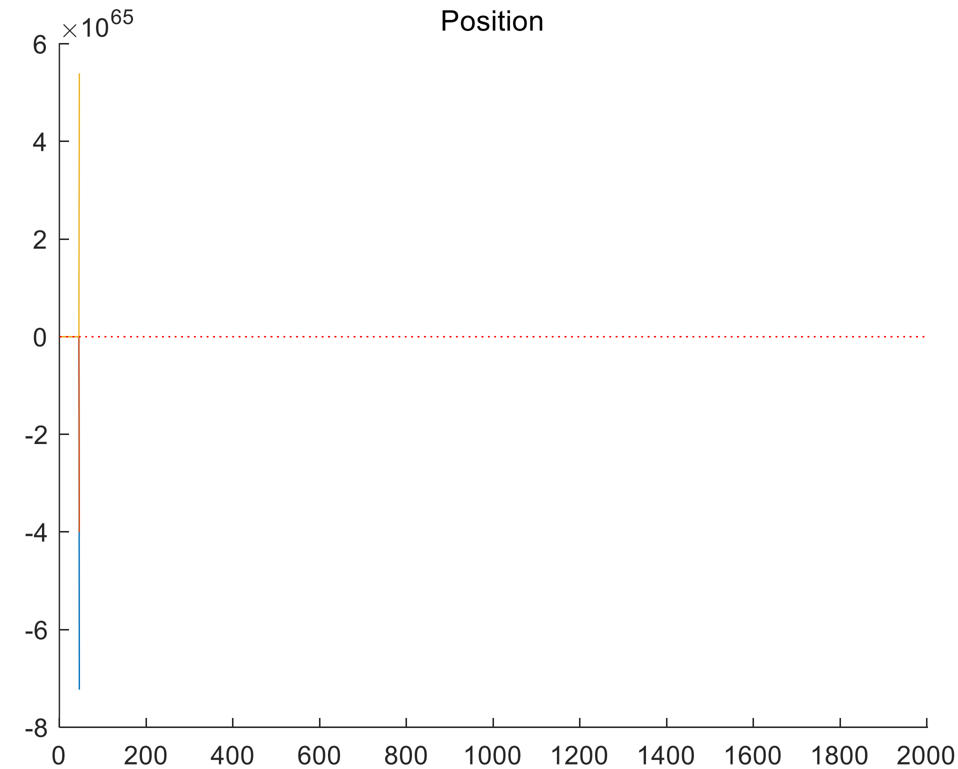


We can see in the first three pictures, estimation error of quaternion is much smaller than the situation in Q1, but estimation errors of position and velocity do not have obvious change. And the covariance of quaternion, velocity and position are smaller than the situation in the Q1.

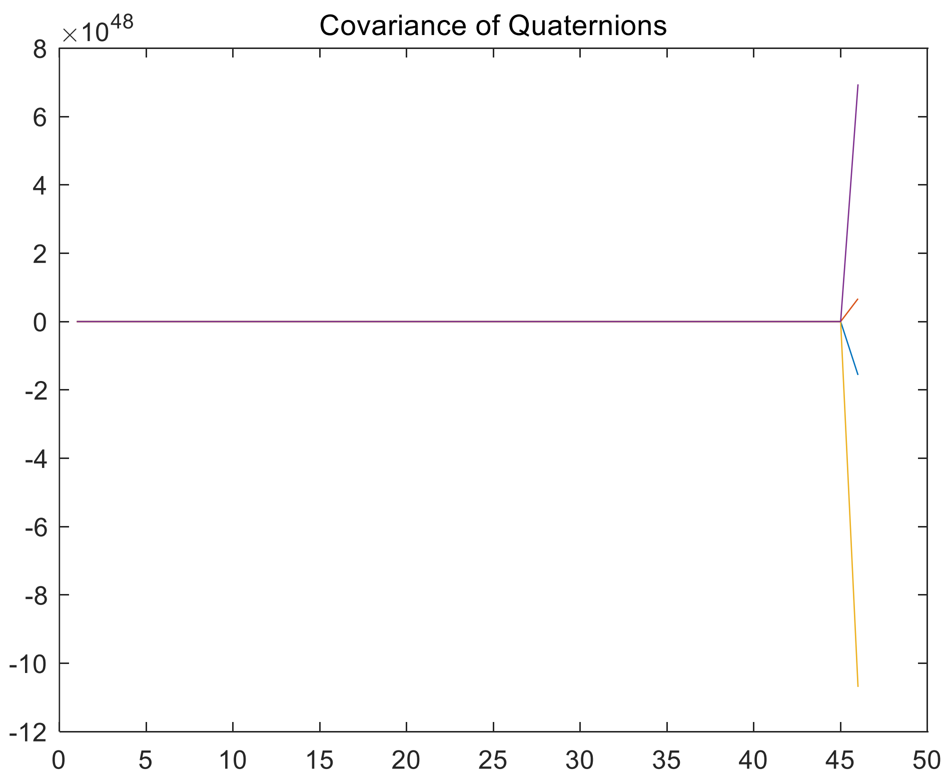
2. $R = [.01\ .01\ .01\ .01\ .01\ .01]$


Quat
×10^47
4
3
2
1
0
-1
-2
-3
-4
0
200
400
600
800
1000
1200
1400
1600
1800
2000

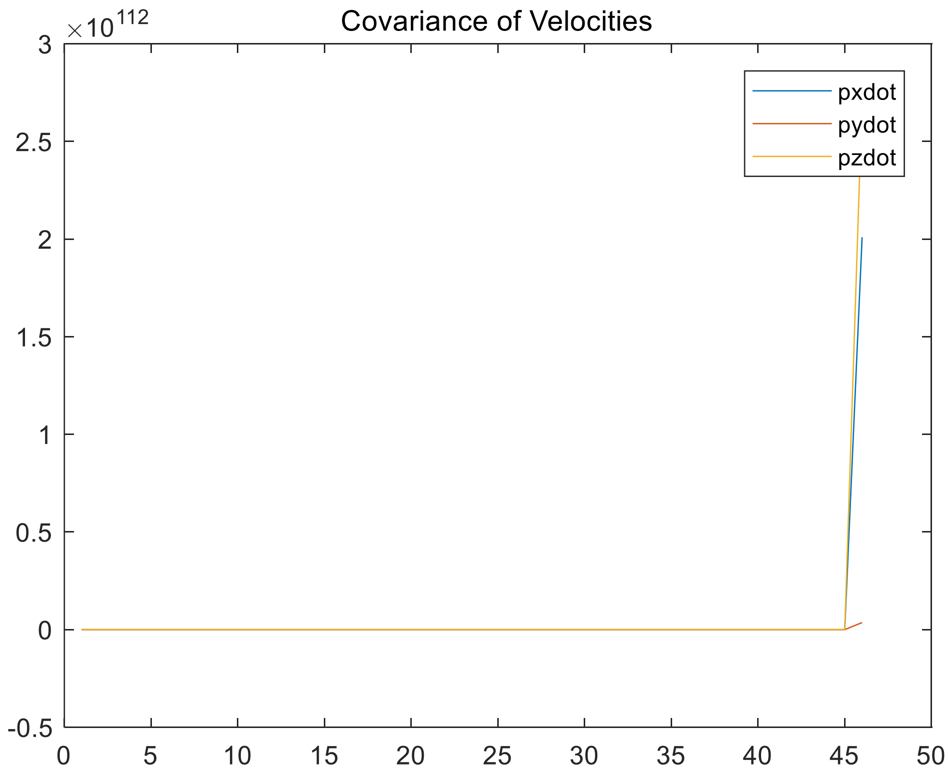

vel x y z
×10^76
6
5
4
3
2
1
0
-1
0
200
400
600
800
1000
1200
1400
1600
1800
2000

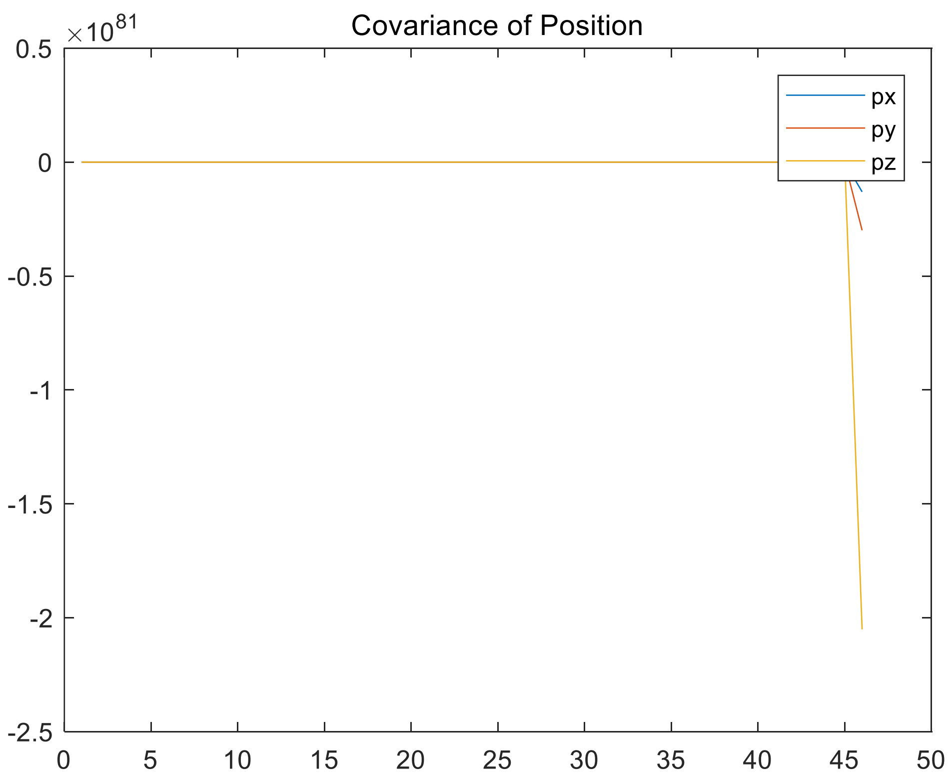


Position
×10^65
6
4
2
0
-2
-4
-6
-8
0
200
400
600
800
1000
1200
1400
1600
1800
2000



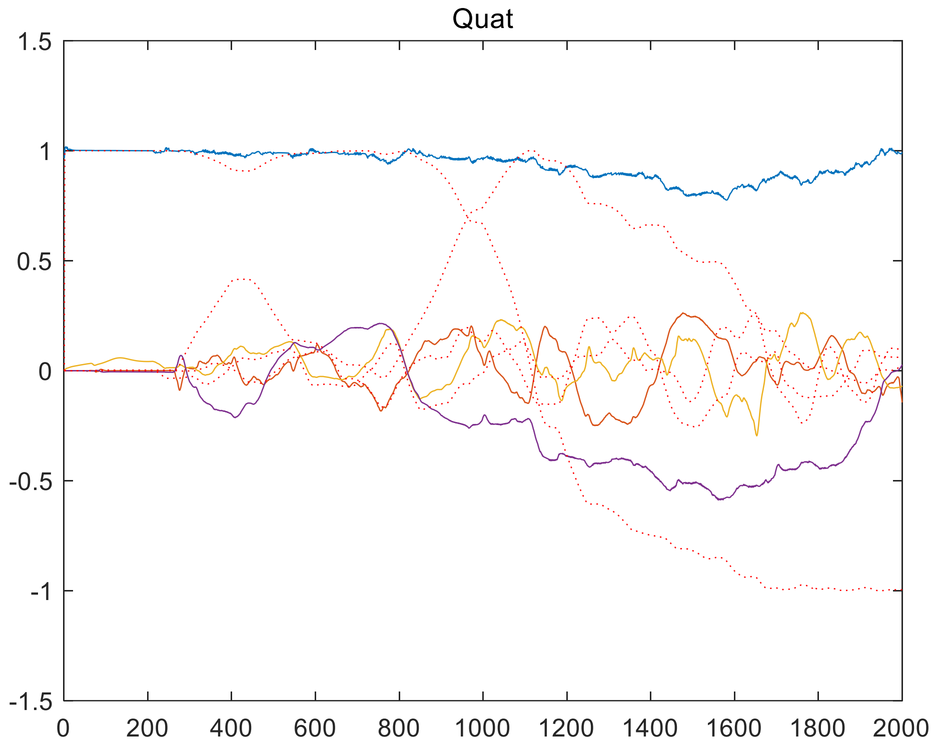
Covariance of Quaternions
×10^48
8
6
4
2
0
-2
-4
-6
-8
-10
-12
0
5
10
15
20
25
30
35
40
45
50

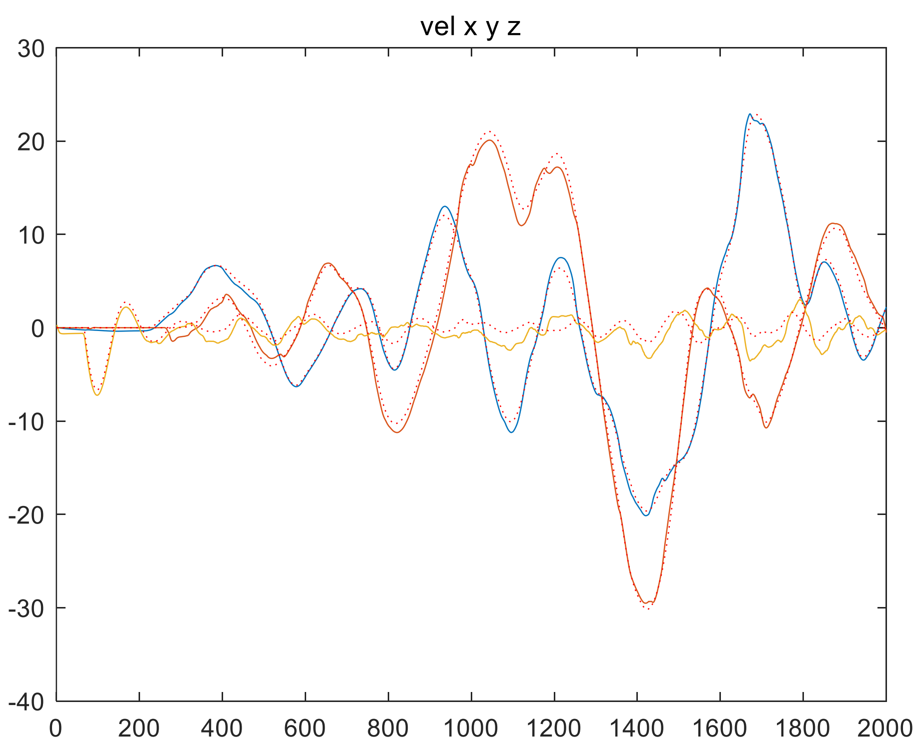

Covariance of Velocities
×10^112
3
2.5
2
1.5
1
0.5
0
-0.5
0
5
10
15
20
25
30
35
40
45
50
pxdot
pydot
pzdot

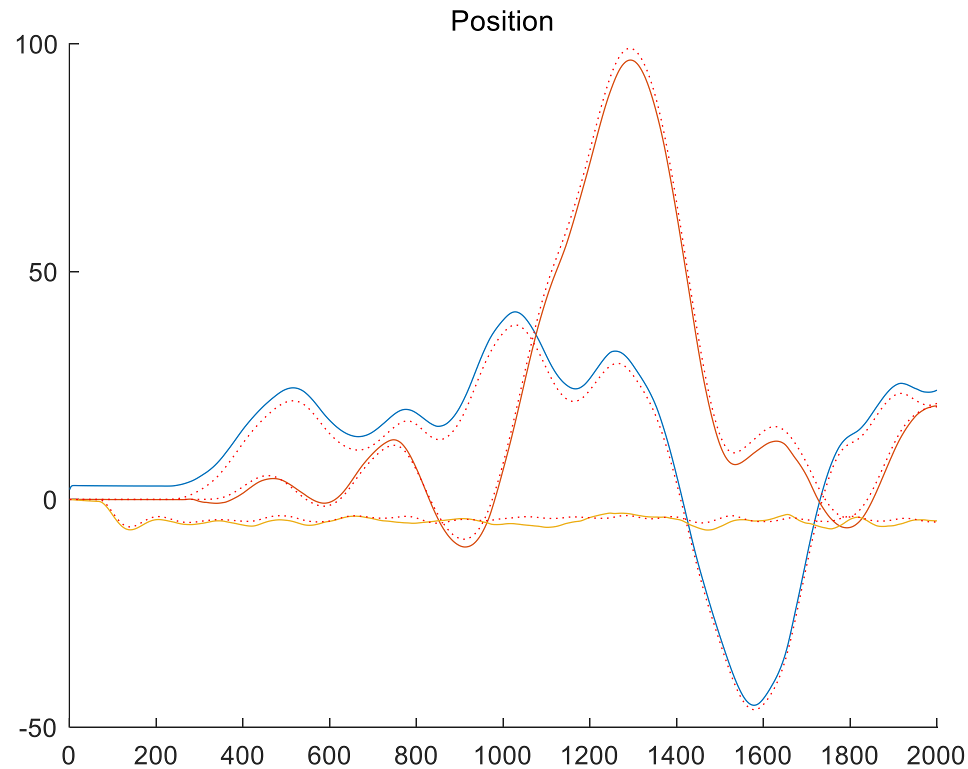

Covariance of Position
×10^81
0.5
0
-0.5
-1
-1.5
-2
-2.5
0
5
10
15
20
25
30
35
40
45
50
px
py
pz

In this situation, the results have the problem of singularity since the R matrix is too small so that it is too closed to the singular value.

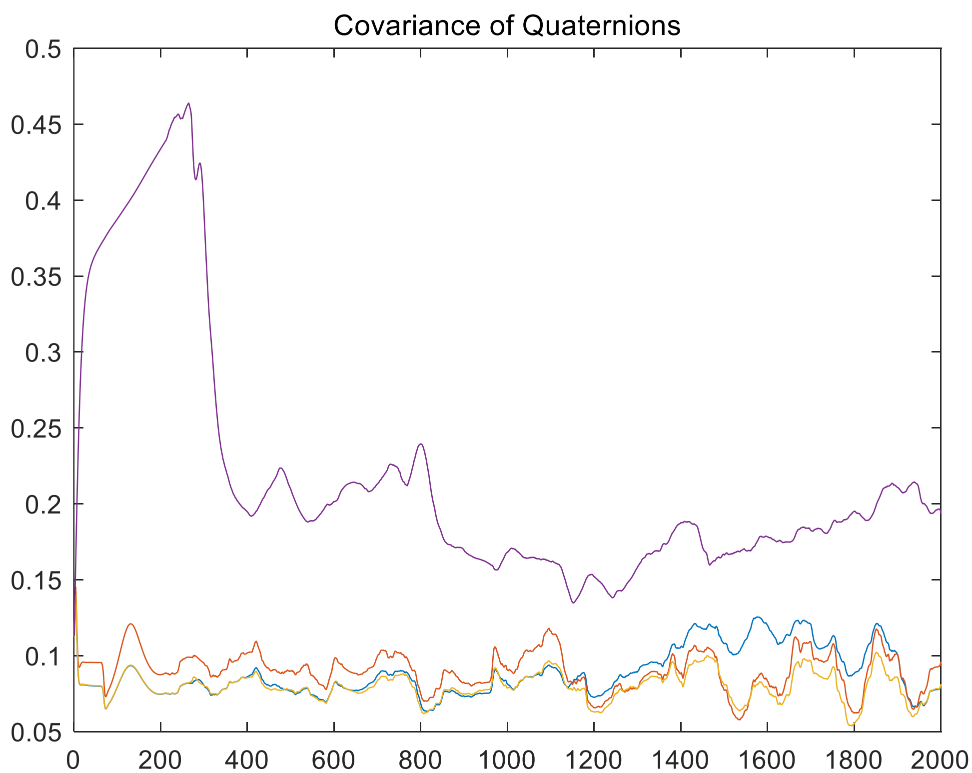
3. $R = [10\ 10\ 10\ 8\ 8\ 8]$



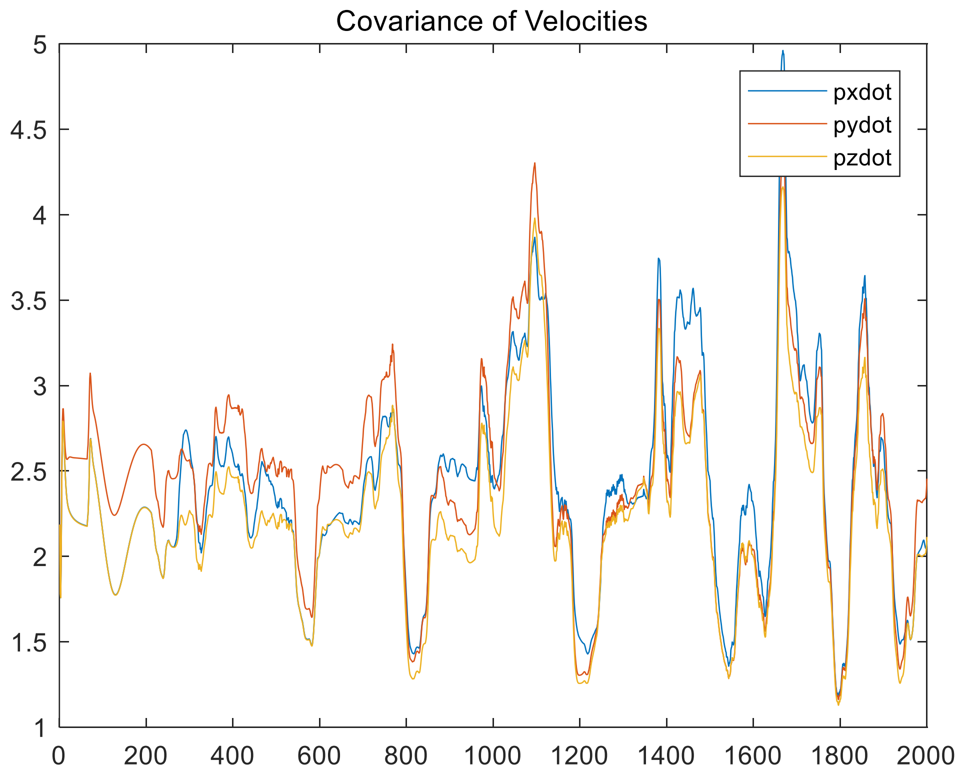


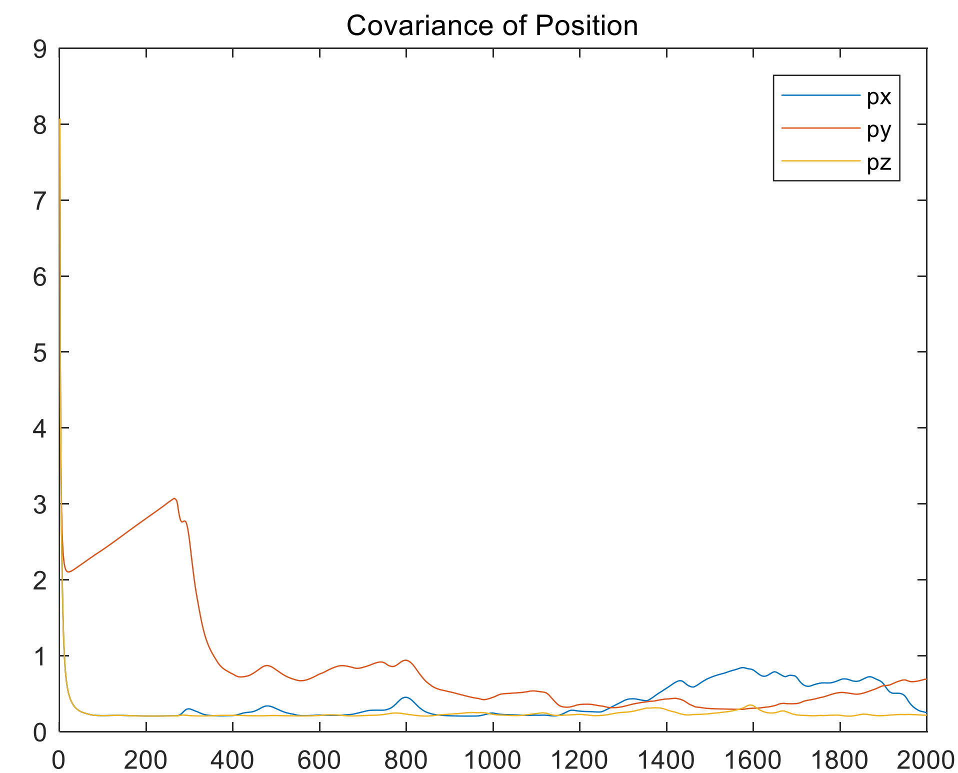


Position
100
50
0
-50
0
200
400
600
800
1000
1200
1400
1600
1800
2000



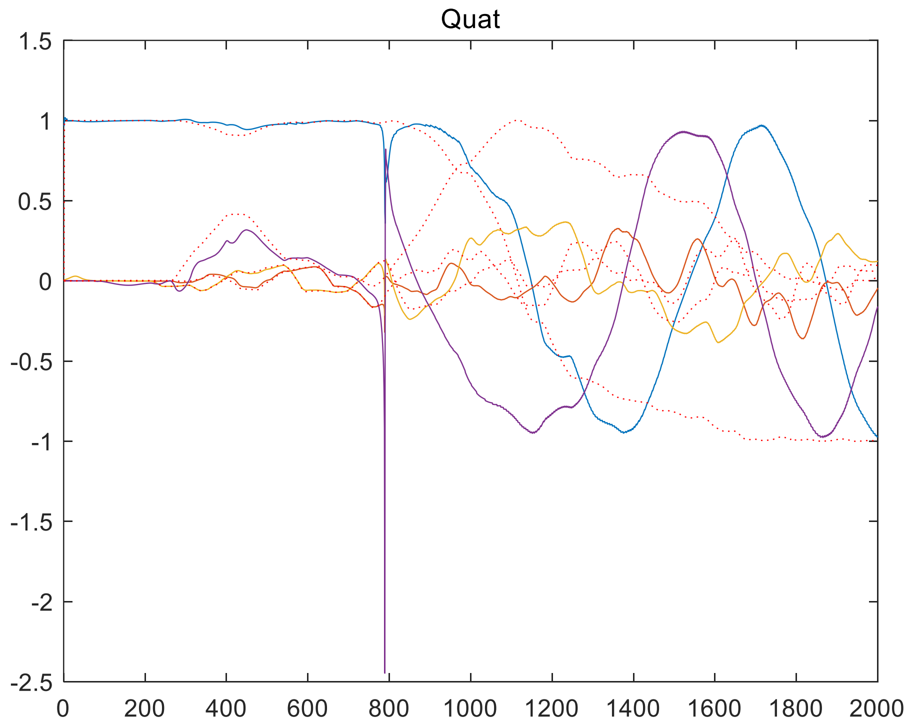
Covariance of Quaternions
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
200
400
600
800
1000
1200
1400
1600
1800
2000

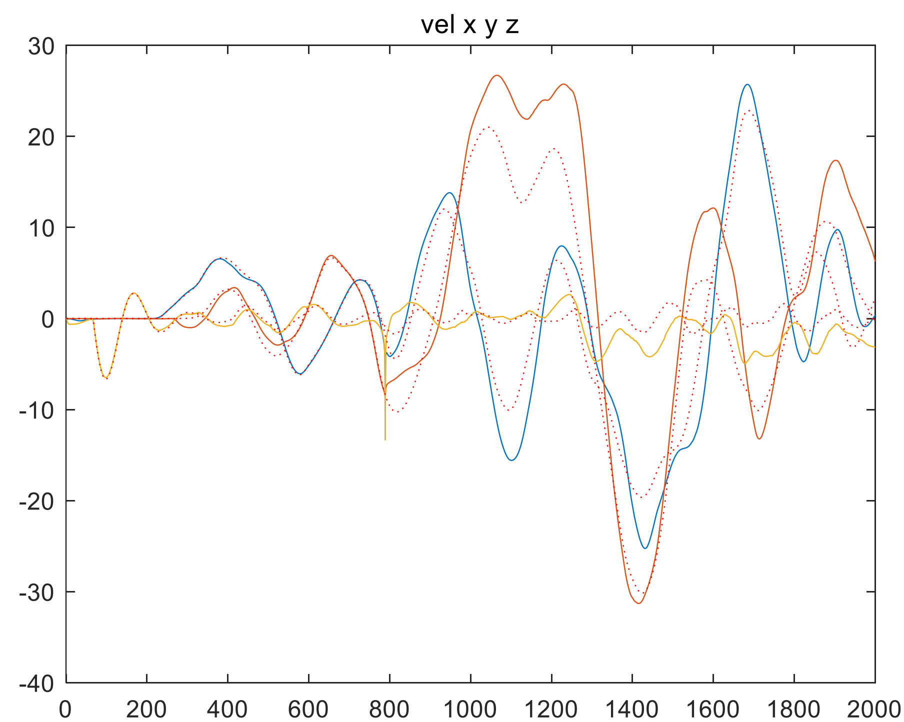

Covariance of Velocities
pxdot
pydot
pzdot
5
4.5
4
3.5
3
2.5
2
1.5
1
0
200
400
600
800
1000
1200
1400
1600
1800
2000

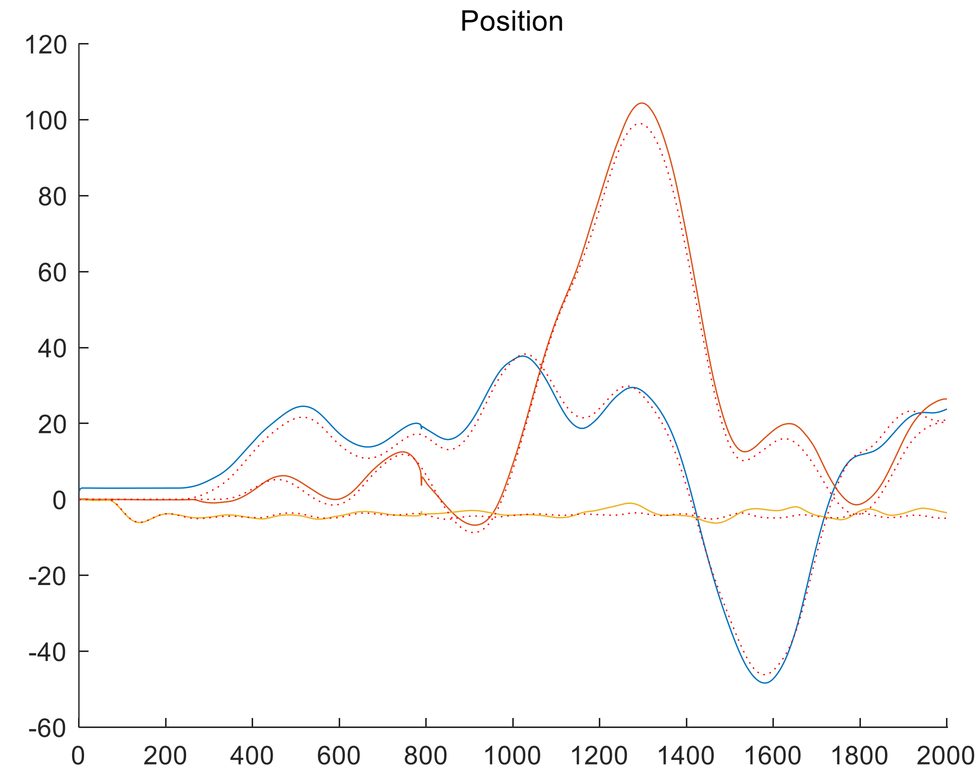

Covariance of Position
px
py
pz
9
8
7
6
5
4
3
2
1
0
0
200
400
600
800
1000
1200
1400
1600
1800
2000

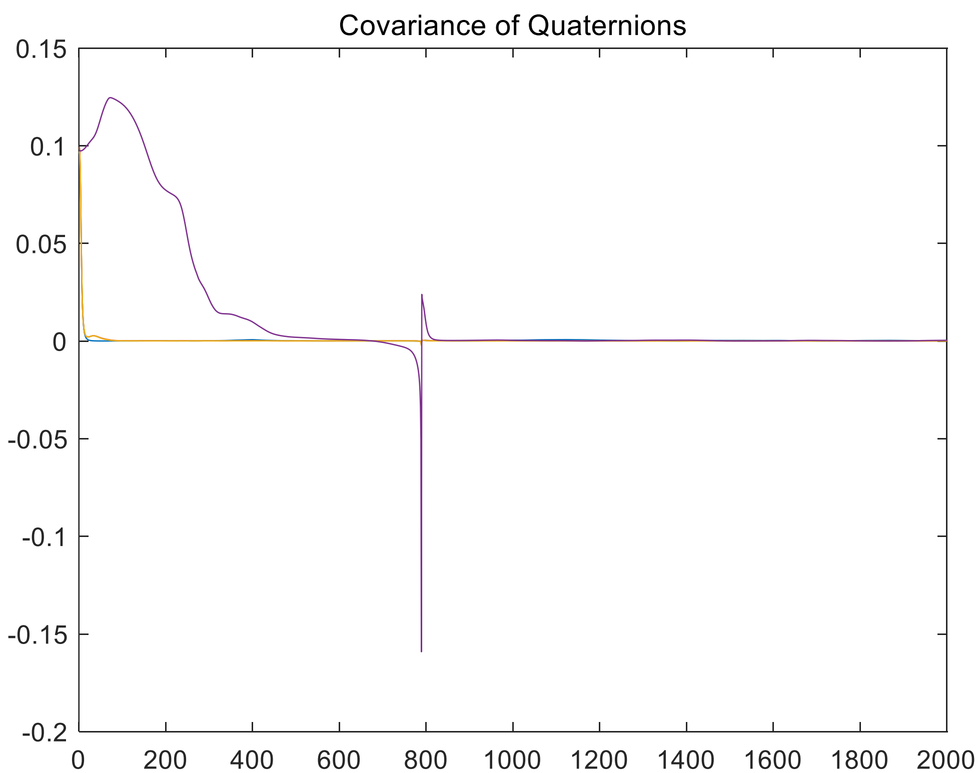
Estimation error of quaternion, position and velocity do not have obvious change. But the covariance of quaternion, velocity and position are little bigger than the situation in the Q1.

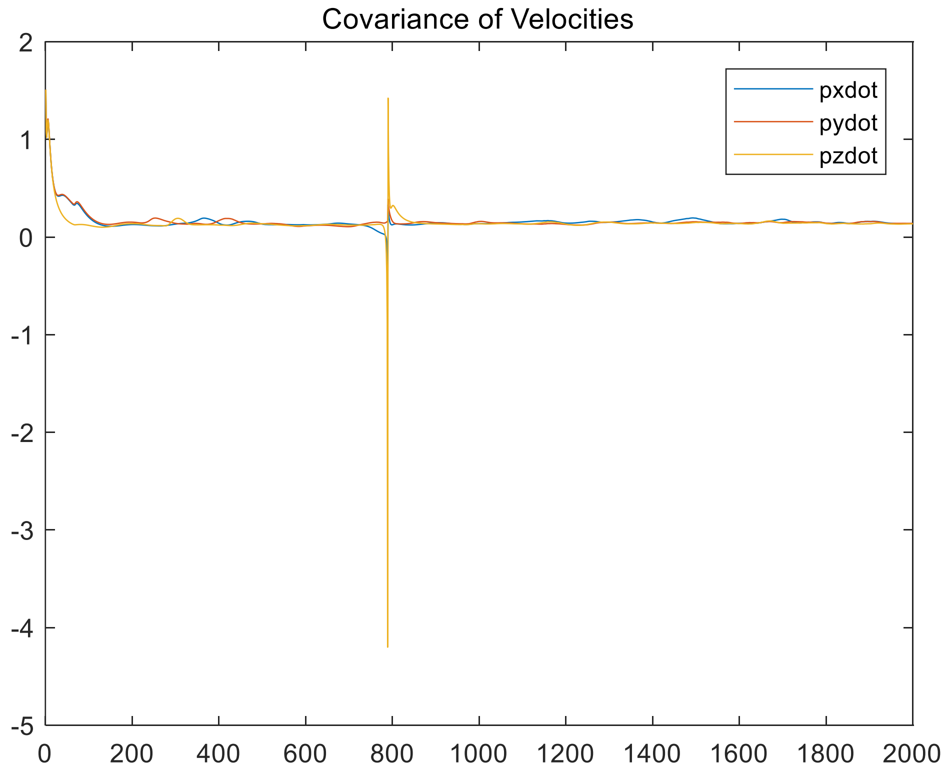
4. $Q = 0.0001 * Q_{original}$

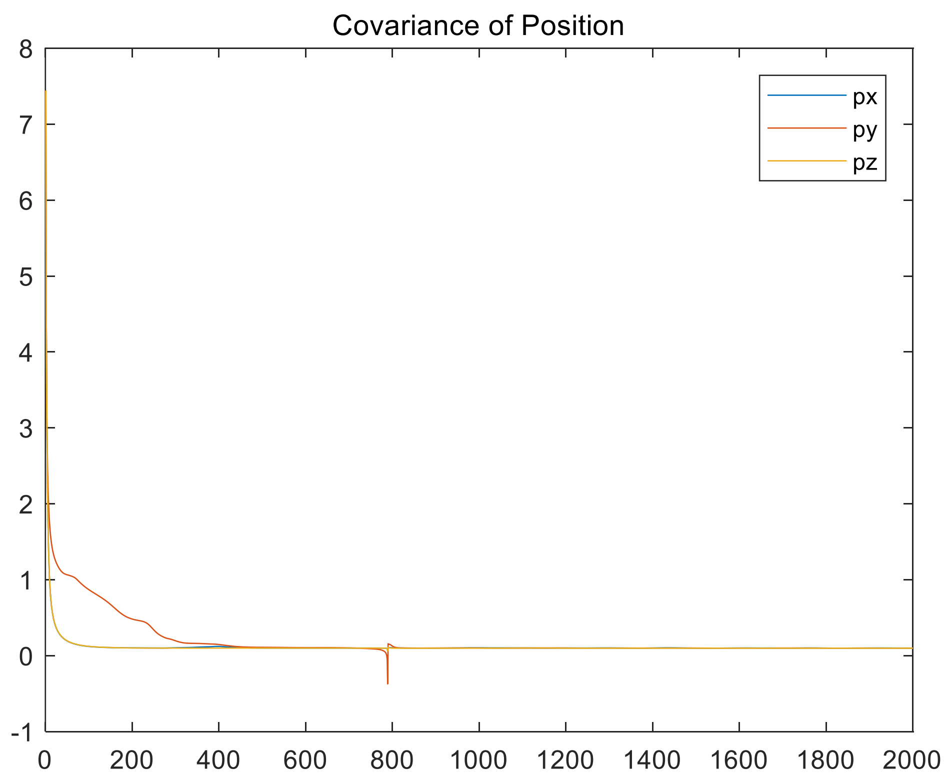





Position
120
100
80
60
40
20
0
-20
-40
-60
0
200
400
600
800
1000
1200
1400
1600
1800
2000

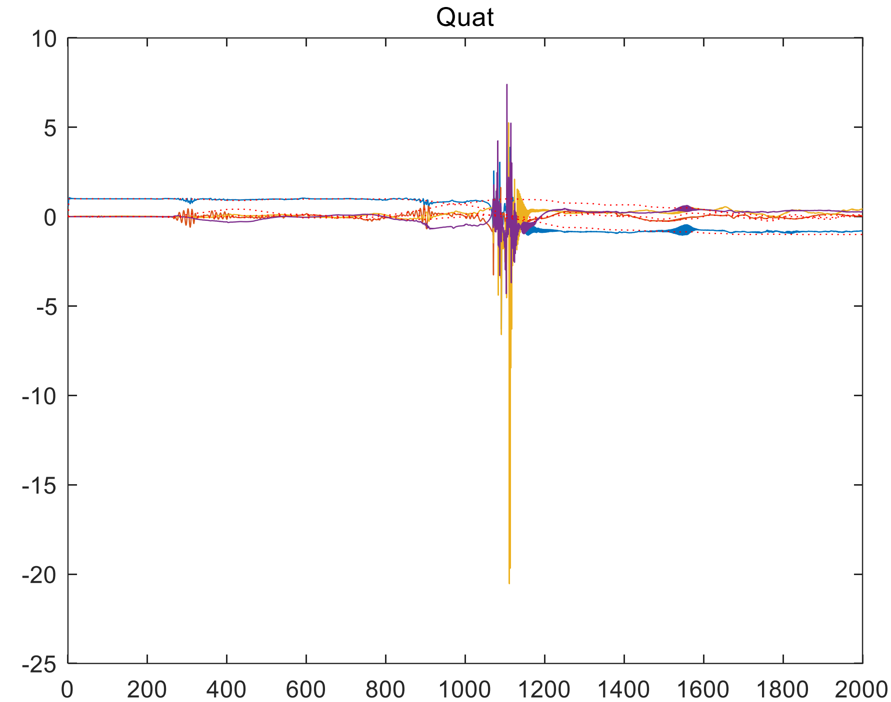

Covariance of Quaternions
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
-0.2
0
200
400
600
800
1000
1200
1400
1600
1800
2000

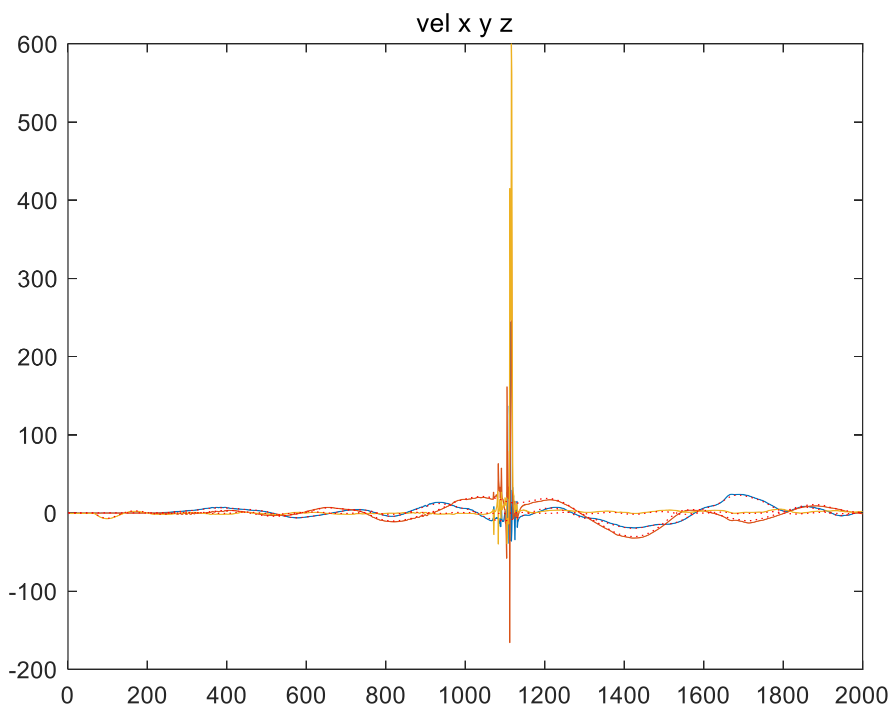

Covariance of Velocities
pxdot
pydot
pzdot
2
1
0
-1
-2
-3
-4
-5
0
200
400
600
800
1000
1200
1400
1600
1800
2000

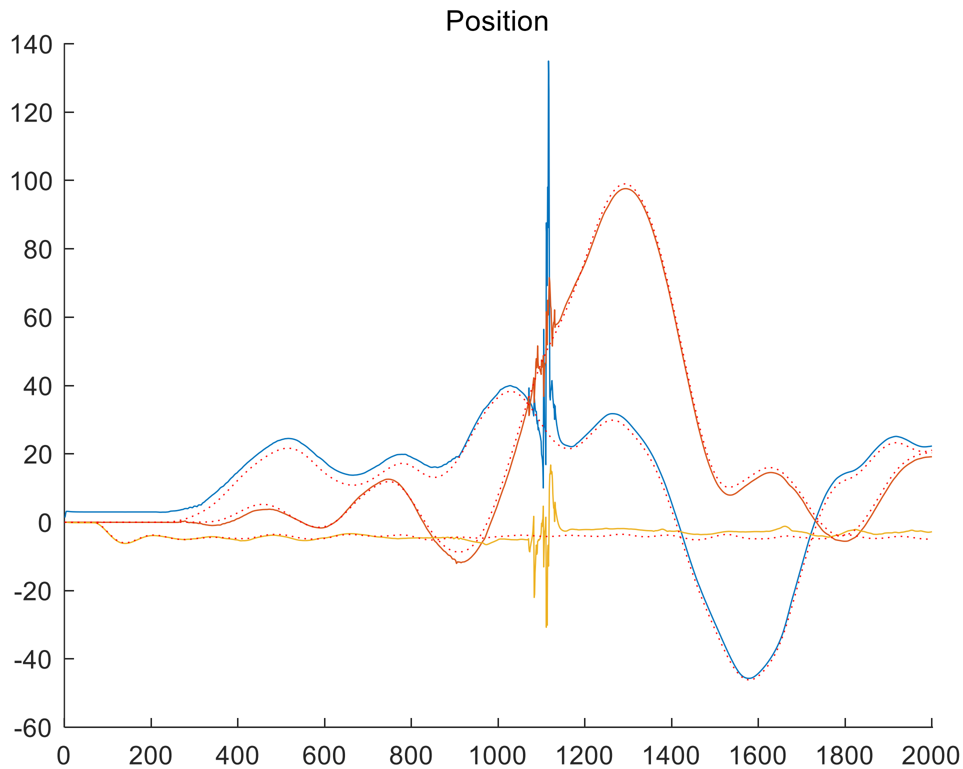

Covariance of Position
px
py
pz
8
7
6
5
4
3
2
1
0
-1
0
200
400
600
800
1000
1200
1400
1600
1800
2000

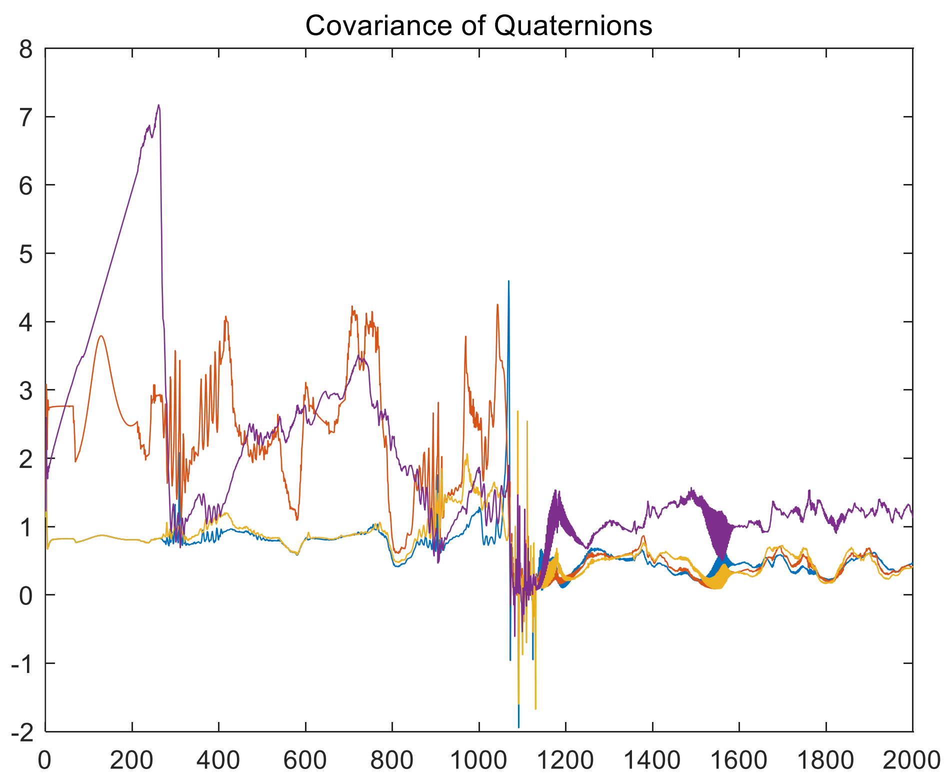
Estimation error of quaternion is smaller than the situation in Q1, and estimation errors of position does not have obvious change, but estimation errors of velocity is a little bigger than the situation in Q1. And the covariance of quaternion, velocity and position are almost zero which are much smaller than the situation in the Q1.

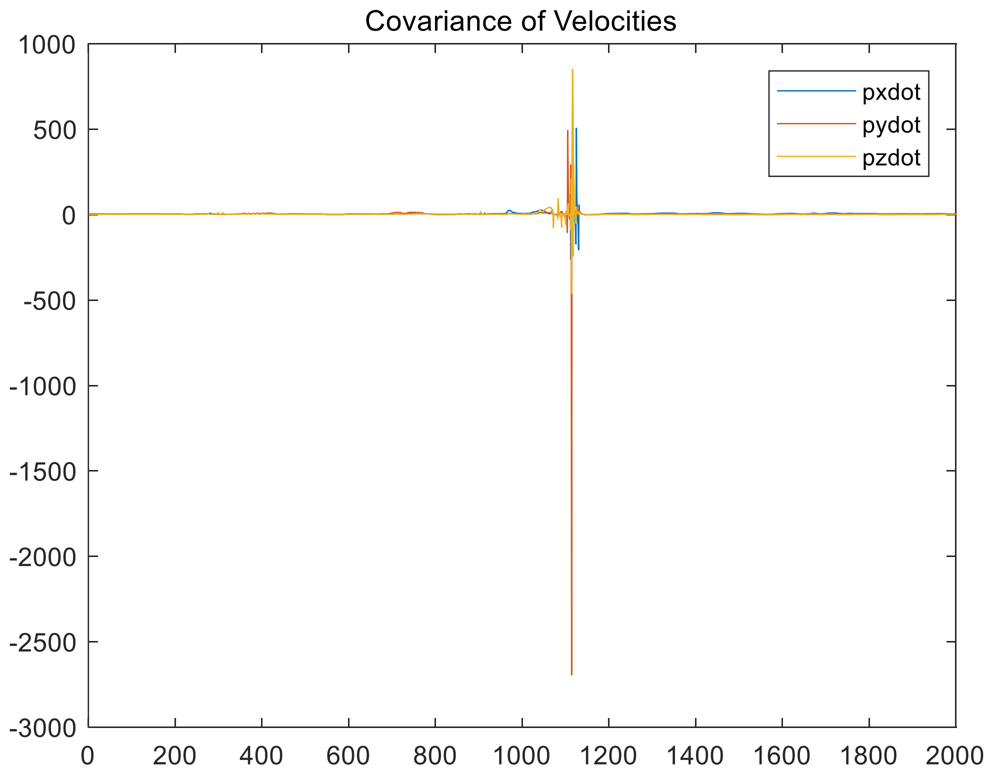
5. $Q = 100 * Q_{original}$

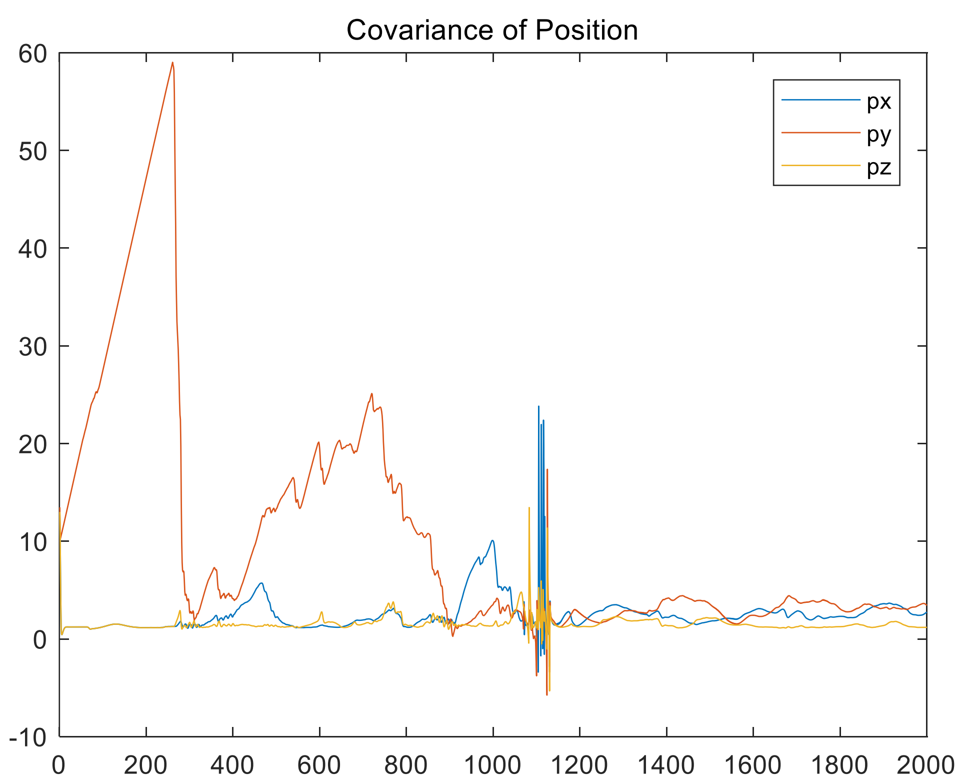





Position
140
120
100
80
60
40
20
0
-20
-40
-60
0
200
400
600
800
1000
1200
1400
1600
1800
2000


Covariance of Quaternions
8
7
6
5
4
3
2
1
0
-1
-2
0
200
400
600
800
1000
1200
1400
1600
1800
2000

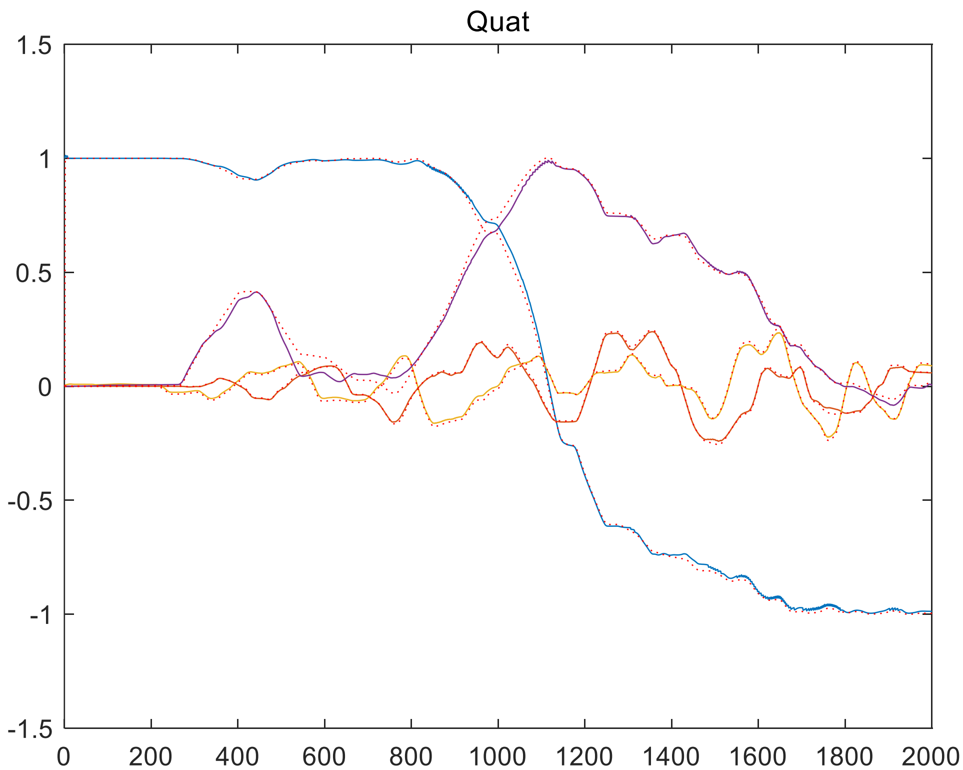

Covariance of Velocities
pxdot
pydot
pzdot
1000
500
0
-500
-1000
-1500
-2000
-2500
-3000
0
200
400
600
800
1000
1200
1400
1600
1800
2000

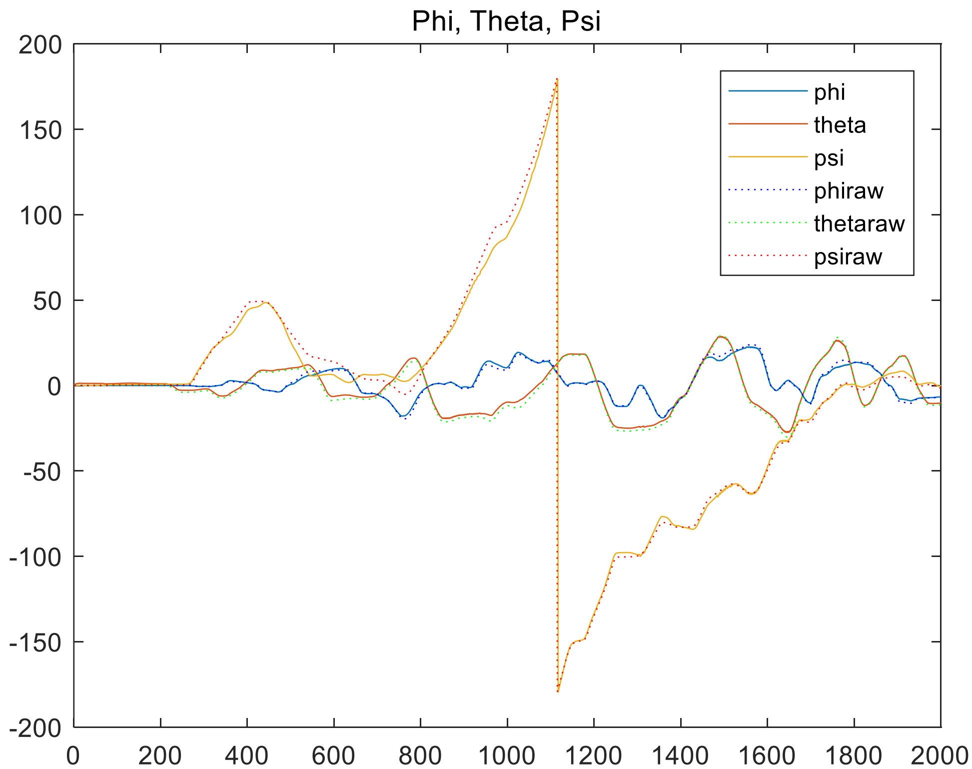

Covariance of Position
px
py
pz
60
50
40
30
20
10
0
-10
0
200
400
600
800
1000
1200
1400
1600
1800
2000

Estimation error of quaternion, position and velocity are much bigger than the situation in the Q1. And the covariance of quaternion, velocity and position are also much bigger than the situation in the Q1.

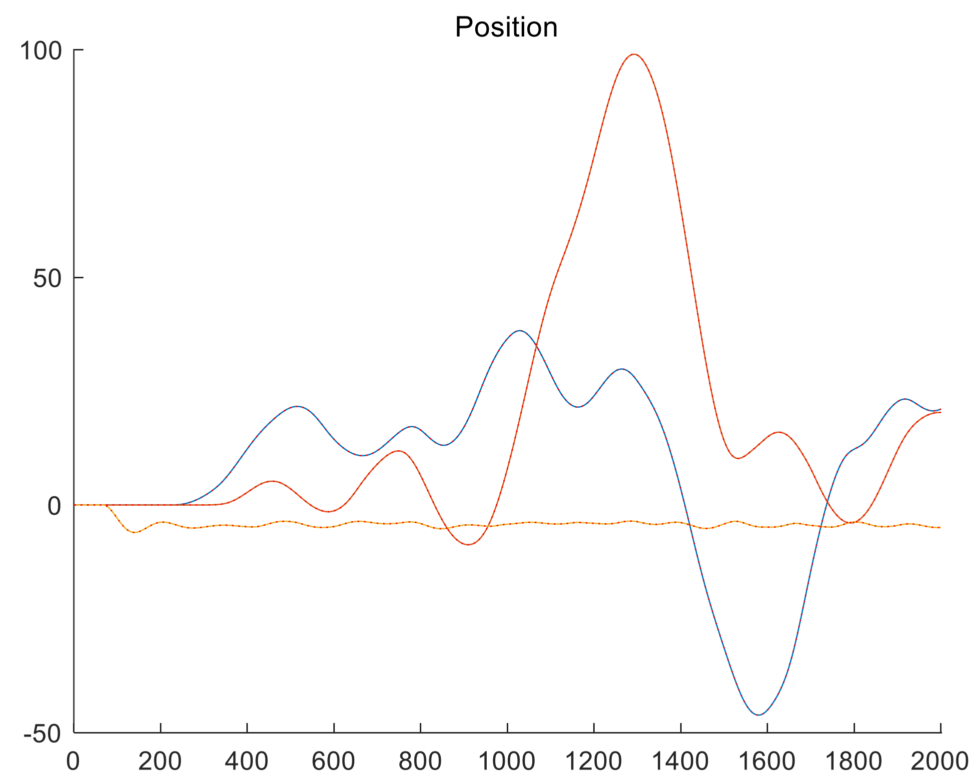
Conclusion: this filter is not robust to very big Q or very small Q.

Q3: Set the R_GPS to zero (I.e. assume the GPS is mounted at the CG), what happens to position estimates and the attitude estimates? Explain why it happens. (10 points)

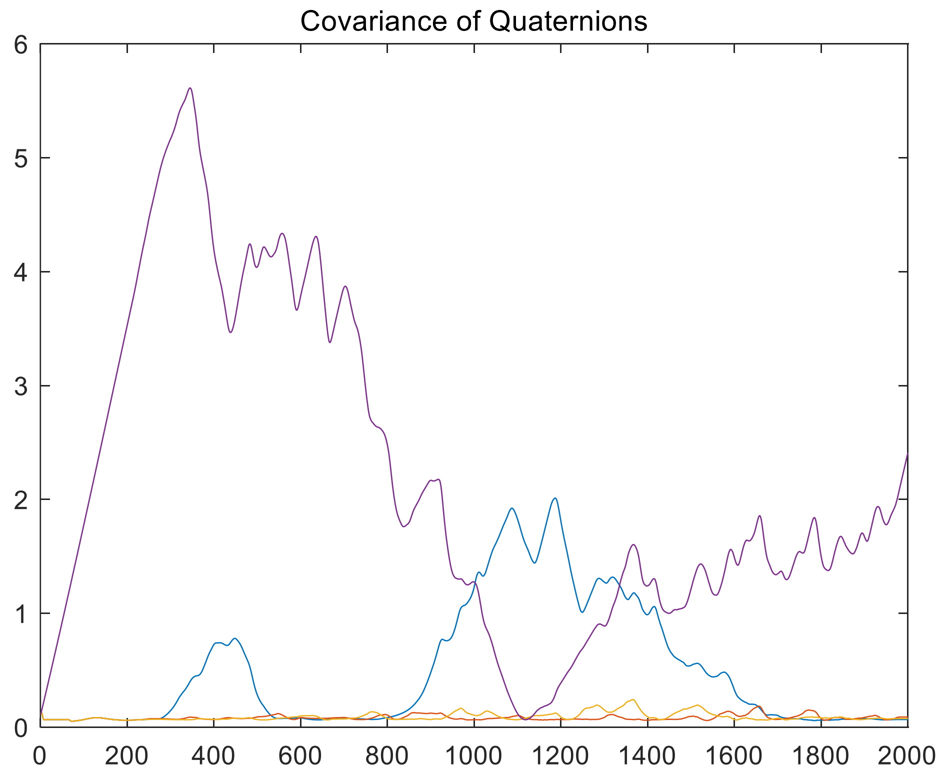




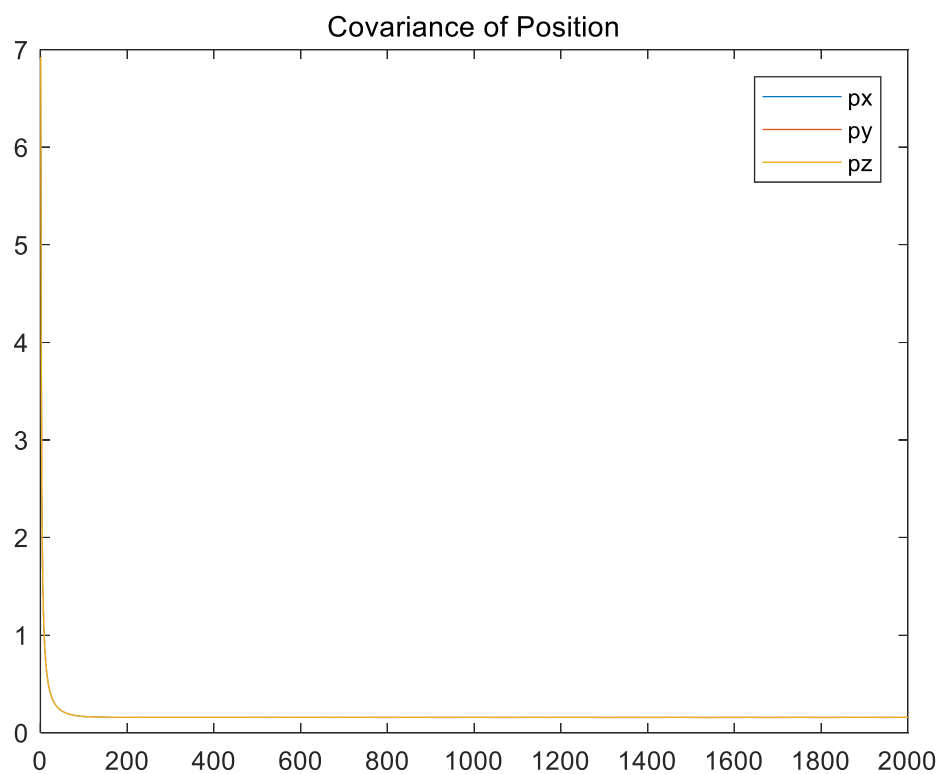
Phi, Theta, Psi
phi
theta
psi
phiraw
thetaraw
psiraw
200
150
100
50
0
-50
-100
-150
-200
0
200
400
600
800
1000
1200
1400
1600
1800
2000



Position
100
50
0
-50
0
200
400
600
800
1000
1200
1400
1600
1800
2000



Covariance of Quaternions
6
5
4
3
2
1
0
0
200
400
600
800
1000
1200
1400
1600
1800
2000



Covariance of Position
7
6
5
4
3
2
1
0
0
200
400
600
800
1000
1200
1400
1600
1800
2000
px
py
pz

In general, if we set r_GPS = [0, 0, 0], it won't have influence on the position estimation, but will have influence on attitude estimation since yaw angle is not observable with accelerometer and gyroscope. So the GPS should not be mounted at CG.

But in this question, if we set r_GPS = [0, 0, 0], the estimation errors of position and quaternion are smaller than the situation in the Q1 as shown in the pictures above. The covariance of position is a little smaller than the situation in the Q1. However, the covariance of quaternion is much bigger than the situation in the Q1 though the estimation error looks much smaller. I do not why this situation happens, but I think there might be some deficiency for improvement in my code.