04-13-22 - HW 5

Wednesday, April 13, 2022 6:15 PM

13.7. We know that when an object reaches terminal velocity, $\dot{x}_2 = 0$

So $\rho_0 e^{(-x_1/k)} \frac{x_2^2}{2x_3} - g = 0 \Rightarrow x_2 = (2g x_3 e^{(x_1/k)}/\rho_0)^{1/2}$

Given altitude = 1mi = 5280 ft = x_1

$k = 22000$

$g = 32.2$

$\rho_0 = 0.0034$

$\dot{x}_3 = \omega_3$ and $E[\omega_3^2] = 0 \therefore x_3 \approx x_{3,0} = 2000$

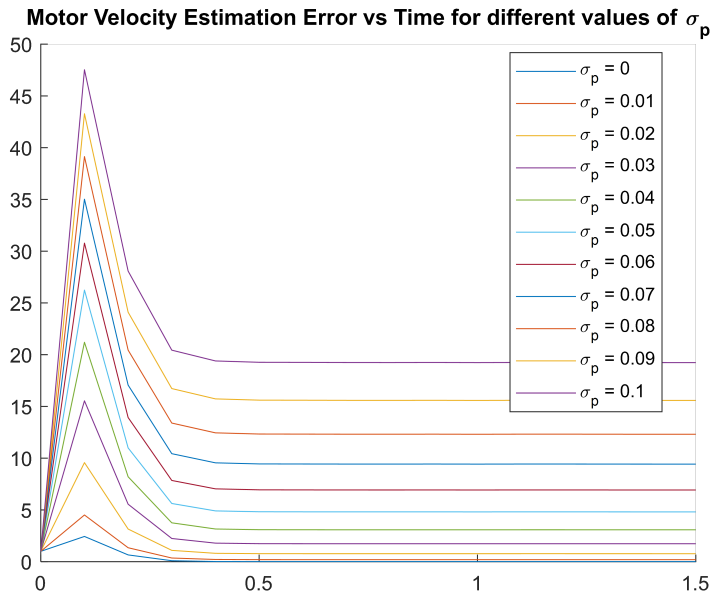
Plugging into previous equation, we get

$x_2 = 6939.590$ ft/s

13.17. Using the constants given in Eg 13.1, a measurement period of 0.1s, measurement noise of 0.1 amps, a hybrid EKF was implemented and and analyzed for control input noise of $\sigma_q = 0 : 0.01 : 0.1$ (Code in Appendix)

The following table and plot shows the performance for different values of σ_q

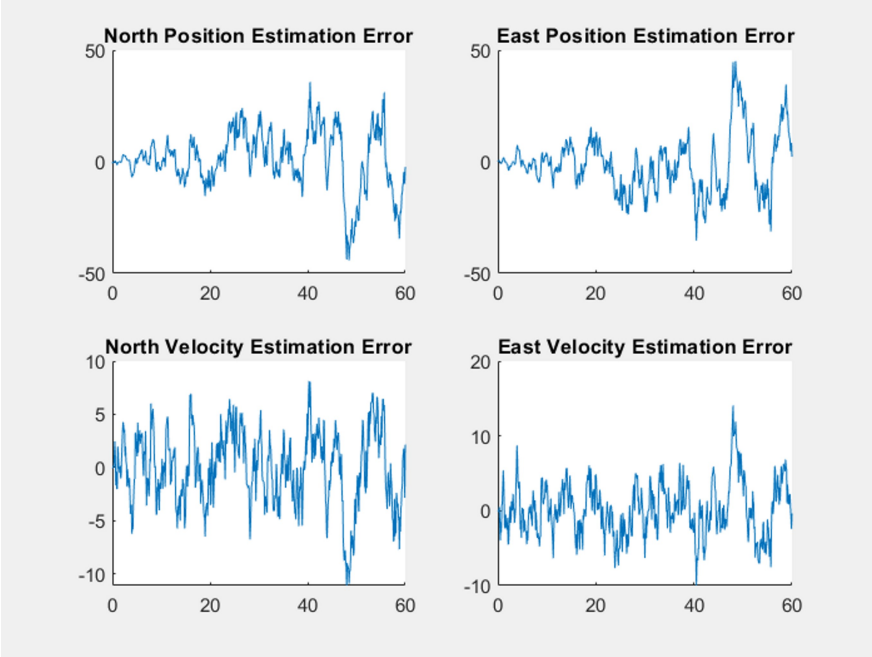
σ_p	Std Dev
0	0.38554
0.01	0.36312
0.02	0.30936
0.03	0.24823
0.04	0.19452
0.05	0.15224
0.06	0.12038
0.07	0.09663
0.08	0.07888
0.09	0.06547
0.1	0.05525



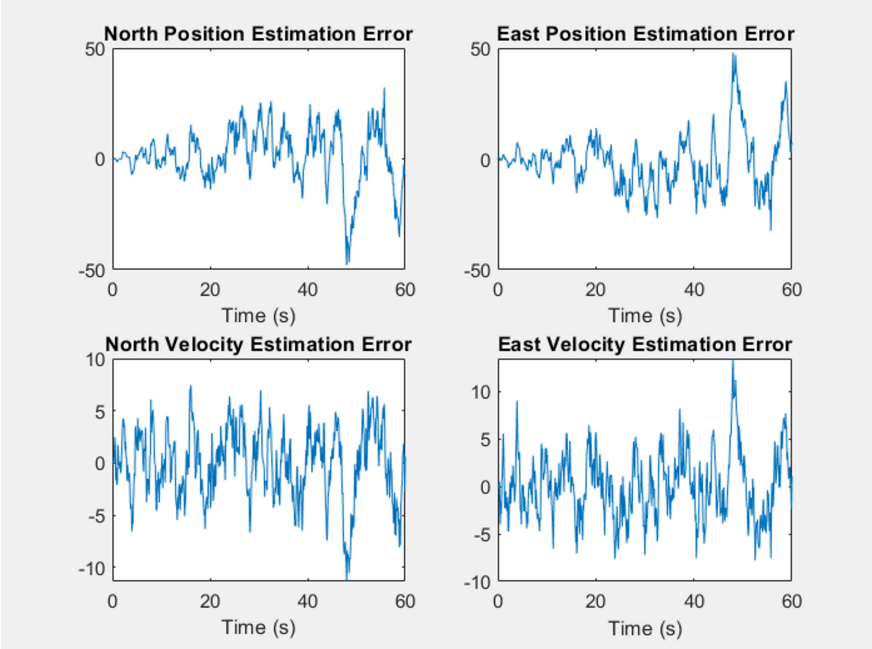
The table shows that the std. dev. of the error decreases as the process noise σ_p increases. The plot shows that the steady state P increases as σ_p increases. Both of these trends are intuitive, since increasing process noise adds a noise floor to P, and this forces the minimum P value to be high, and forces the filter to consider new measurements, which causes the filter to move away from the dynamics and thus increases uncertainty.

14.14. Designed an EKF and UKF (code in Appendix)

EKF results:



UKF results:



To make comparisons more accurate, both filters were initialized with the same seed.

The plots show that the filters do not converge to a steady state value. Position errors are within 50, and velocity errors are within 10. We can also note that there is not a noticeable

difference between the UKF and EKF performance. Since the system is not very nonlinear, the UKF does not give an advantage over EKF.

15.15. a. A particle filter was implemented for the system in Eq 15.1 (code in Appendix) and was analyzed for different values of N. The following was the output from MATLAB:

```
N = 10
Kalman filter RMS error = 17.3186
Particle filter RMS error = 6.6167

N=100
Kalman filter RMS error = 13.1397
Particle filter RMS error = 3.2323

N=1000
Kalman filter RMS error = 13.5754
Particle filter RMS error = 3.0222
```

It can be seen that the particle filter consistently performs better than the EKF. As N increases, the particle filter fits the data better until it converges around 3.

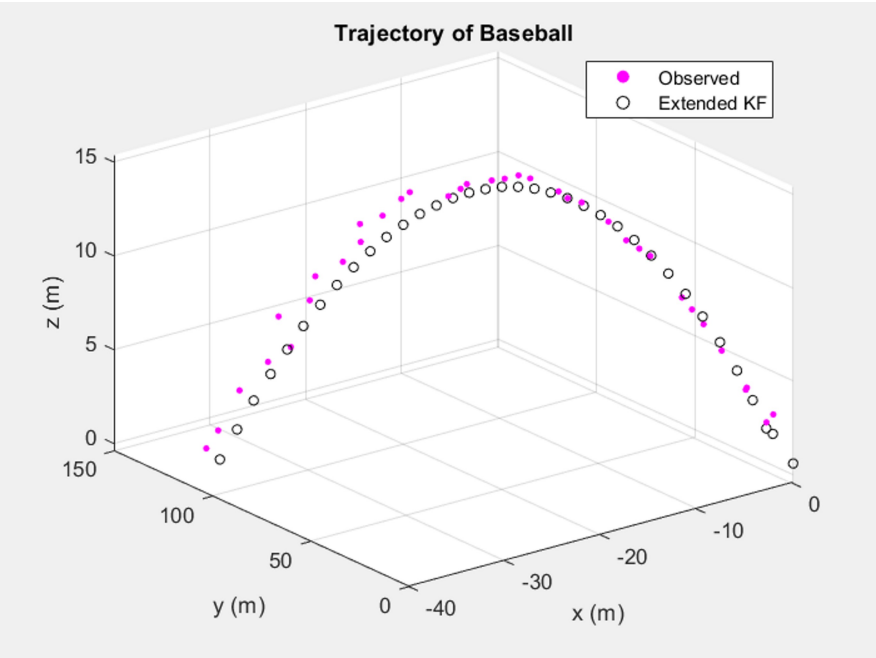
b. The effect of different values of Q was tested:

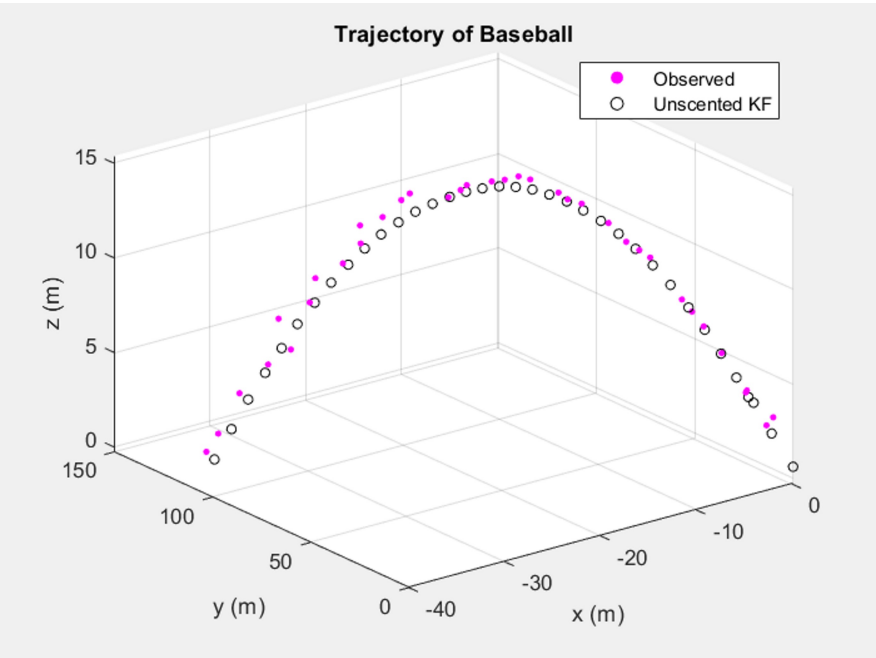
```
Q = 0.1
Kalman filter RMS error = 13.6732
Particle filter RMS error = 1.486

Q = 1
Kalman filter RMS error = 14.6671
Particle filter RMS error = 3.3447

Q = 10
Kalman filter RMS error = 25.1239
Particle filter RMS error = 5.0115
```

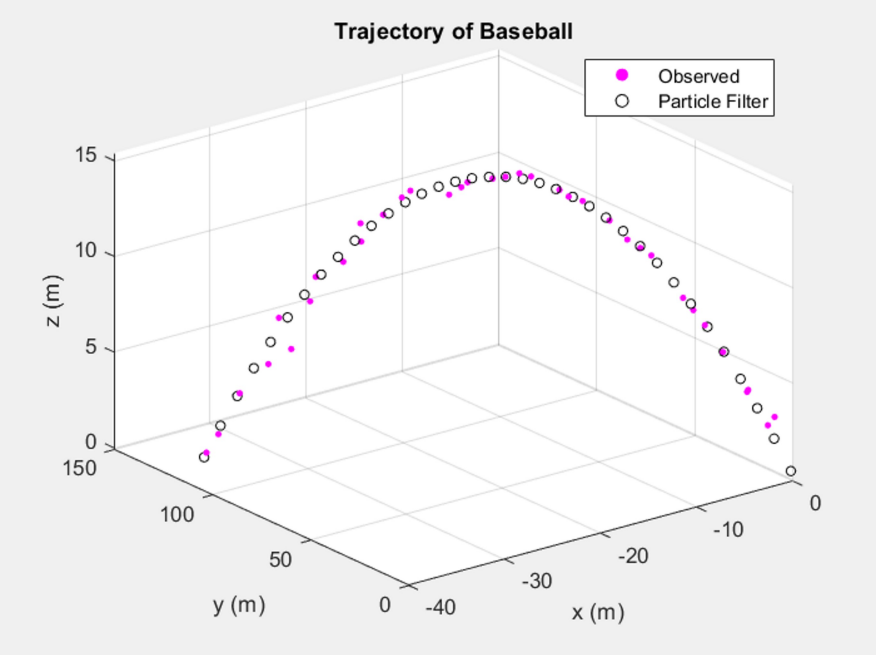
As Q increases, the RMS increases. This is intuitive because

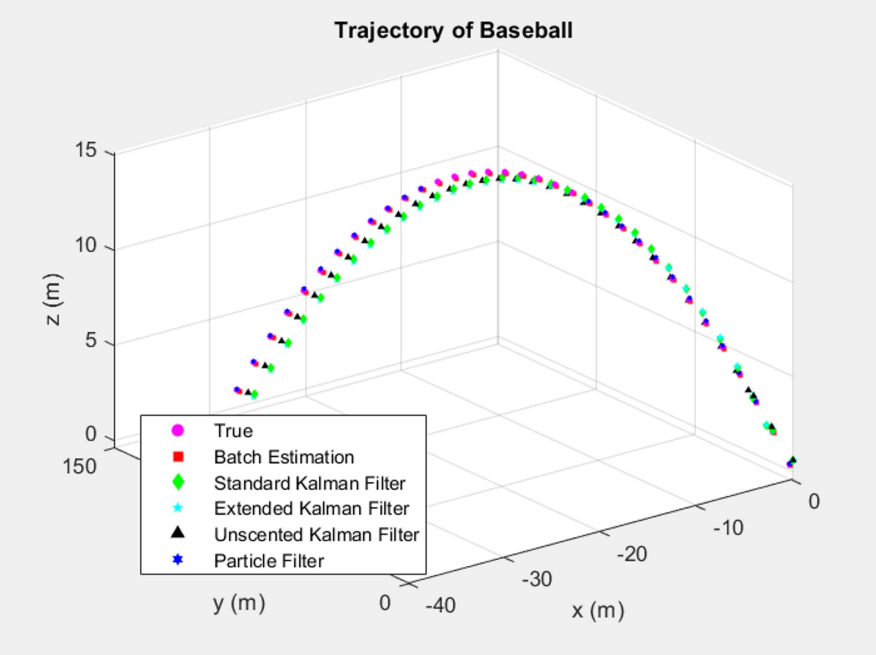
as Q increases, the filter begins to ignore the dynamics more and pays more attention to the measurements. This causes the filter to move away from the true states.

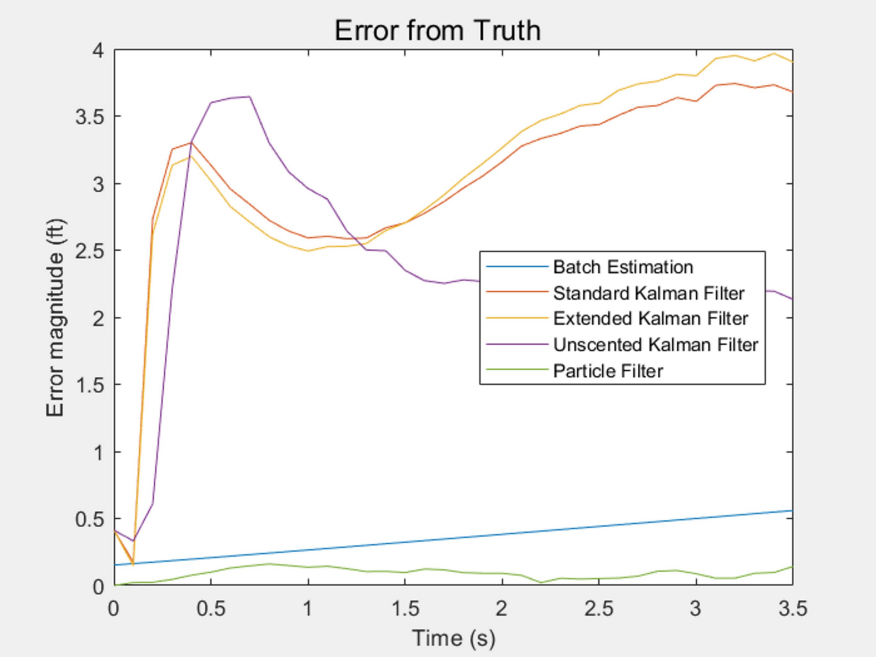
2. Code was written to implement an EKF, a UKF, and a particle filter. All of the filters compared with no added process noise.






Trajectory of Baseball
Observed
Particle Filter
z (m)
15
10
5
0
150
100
50
0
y (m)
-40
-30
-20
-10
0
x (m)


Trajectory of Baseball
True
Batch Estimation
Standard Kalman Filter
Extended Kalman Filter
Unscented Kalman Filter
Particle Filter
z (m)
15
10
5
0
150
y (m)
0
-40
-30
-20
-10
0
x (m)



We can see that the batch filter performs well, as it gets very close to the initial state. However, this approach is not good for real-time systems.

The extended KF does not perform much better than the standard KF, but the UKF performs much better, since it addresses the non-linearity of the system. The particle filter also performs very well.

The std dev of the estimation error of position by the various filters is as follows:

```
Std Dev of Estimation Error of position with batch estimation   0.122937
Std Dev of Estimation Error of position with standard KF        0.77594
Std Dev of Estimation Error of position with EKF                0.84945
Std Dev of Estimation Error of position with UKF                0.744295
Std Dev of Estimation Error of position with particle filter    0.040796
```