Optimal Observers: Kalman Filters

Suppose you have a system with disturbances:

$$sX = AX + BU + Fv \qquad v = \eta(0, V^2)$$

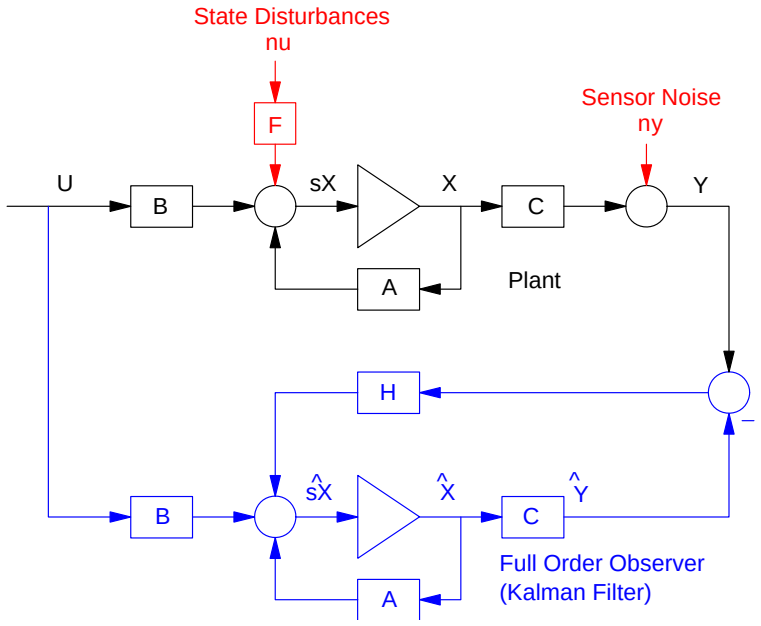
$$Y = CX + w \qquad w = \eta(0, W^2)$$

What is the "best" estimate of the stated given such disturbances?

Here,

- Fv models disturbances on the states (sun shining on an apartment, vibrations, etc)
- w models the sensor noise (no sensor is perfect)

Using a full-order observer, you can estimate the states as follows:



$$\begin{bmatrix} sX \\ s\hat{X} \end{bmatrix} = \begin{bmatrix} A & 0 \\ HC & A-HC \end{bmatrix} \begin{bmatrix} X \\ \hat{X} \end{bmatrix} + \begin{bmatrix} B \\ B \end{bmatrix} U + \begin{bmatrix} F \\ 0 \end{bmatrix} \eta_x + \begin{bmatrix} 0 \\ H \end{bmatrix} \eta_y$$

or

$$\begin{bmatrix} sX \\ s\hat{X} \end{bmatrix} = \begin{bmatrix} A & 0 \\ HC & A-HC \end{bmatrix} \begin{bmatrix} X \\ \hat{X} \end{bmatrix} + \begin{bmatrix} B & F & 0 \\ B & 0 & H \end{bmatrix} \begin{bmatrix} U \\ \eta_x \\ \eta_y \end{bmatrix}$$

However, the observer gains, H, should reflect the amount of noise on the system:

- If the sensor noise is zero, the observer gain, H, should be large. Essentially, with a perfect measurement, you can determine all the states exactly with the plant model (i.e. with the observer).
- If the state-disturbances are zero, however, the observer gain, H, should be zero. With a perfect model and perfect knowledge of the inputs, you can determine the states precisely.

If you have both state disturbances and sensor noise, however, what is the optimal trade-off between these extremes? What is the optimal observer gain, H, which minimizes the error in the state estimate?

If you define "best" as minimizing the variance of the state error

E = X - X^

the solution is a LQR observer where

$$Q = (FV)(FV)^T = F \cdot V^2 \cdot F^T$$

$$R = WW^T = W^2$$

An LQR observer designed with this value of Q and R is termed a *Kalman Filter*. (It's just a full-order observer with a specific feedback gain, H.)

Example: 4th Order Heat Equation

Suppose you have a 4-order heat equation with a large disturbance at the input.

$$Fv = Bv = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} v \qquad v \sim N(0, 1^2)$$

Determine the optimal observer with large sensor nosie to small sensor noise:

$w \sim N(0, W^2)$ sensor noise

Also plot the actual and estimated state for sensor noise of

- $W = 0.01$ Good sensors: noise 100x less than disturabances
- $W = 0.1$
- $W = 1$ Noisy Sensors

Note on the following simulations: The state disturbance should be applied to the plant. It shows better if it is applied to the observer, however (as is done in the following simulations). This results in the target (actual state) being a clean line - making the error in the state estimate easier to see.

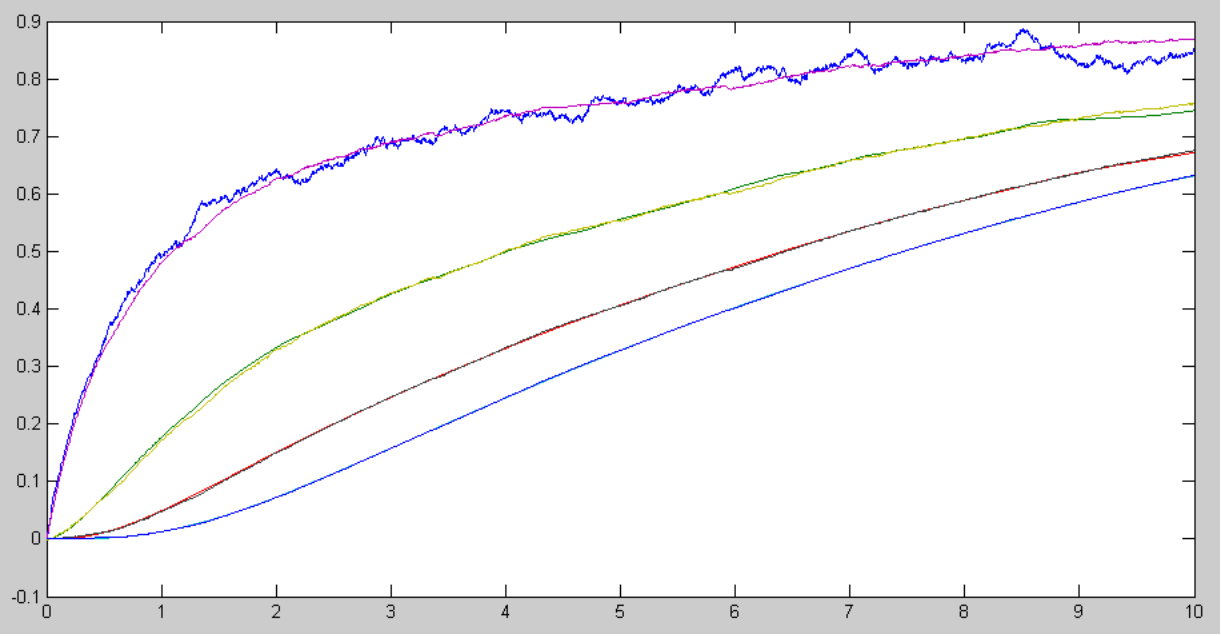
Case 1: Good Sensors (small noise)

- W = 0.01
- V = 1

```
W = 0.01;      % sensor noise
V = 1;         % state disturbance (on U)
Q = F*V*V*F';
R = W*W;
H = lqr(A',  C',  Q,  R)'
A8 = [A, zeros(4,4) ;   H*C,  A - H*C ];
B8 = [B  F  zeros(4,1) ;  B   zeros(4,1)   H];
```

```
     Ref         Nx         Ny
  1.0000    1.0000          0
       0         0          0
       0         0          0
       0         0          0
  1.0000         0    25.5559
       0         0    18.4321   H shows up here
       0         0     8.0233
       0         0     3.1287
```

```
t = [0:0.001:10]';
N = size(t);
X0 = zeros(8,1);
U = [ ones(N),  randn(N)*V,  randn(N)*W ];
C8 = eye(8,8);
D8 = zeros(8,3);
y = step3(A8, B8, C8, D8, t, X0, U );
plot(t,y)
```



States and State Estimates with State and Sensor Noise: $W \sim N(0, 0.01^2)$, $V \sim N(0, 1^2)$

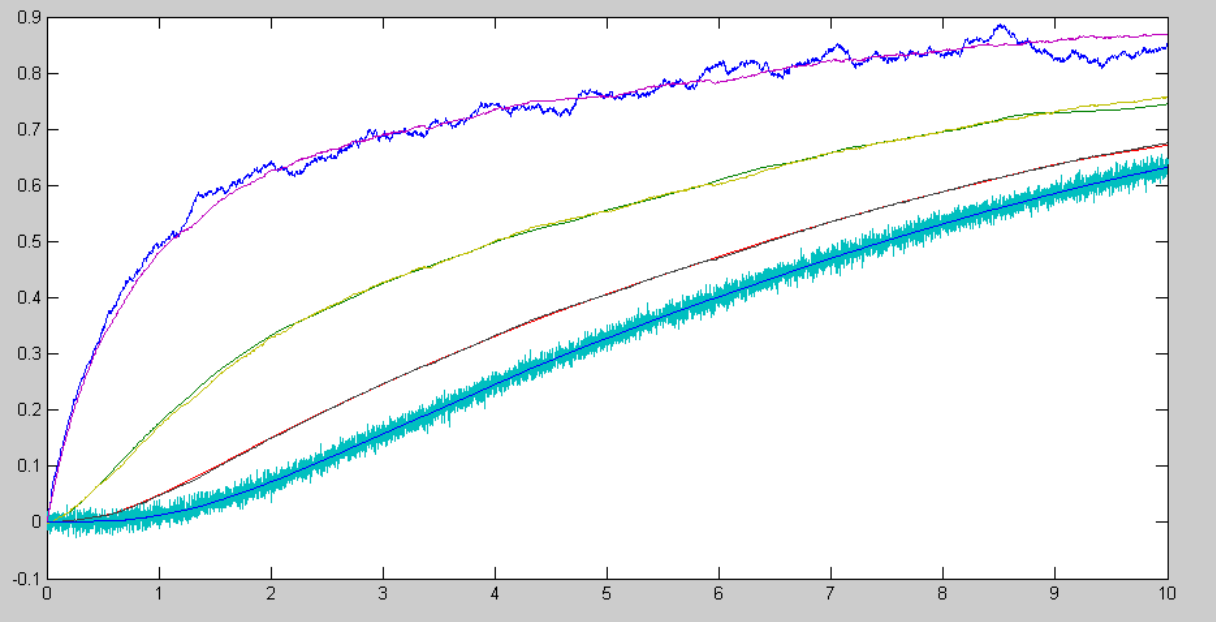
If you change D8, you can see the measured output (i.e. with noise)

```
>> D8(4,3) = 1

D8 =

     0     0     0
     0     0     0
     0     0     0
     0     0     1     sensor noise shows up on measurement of X4 (y)
     0     0     0
     0     0     0
     0     0     0
     0     0     0

>> y = step3(A8, B8, C8, D8, t, X0, U );
>> plot(t,y)
```



States and State Estimates.

Note that even though you measure X4, you might want to use its estimate instead. The estimate has filtered out the noise.

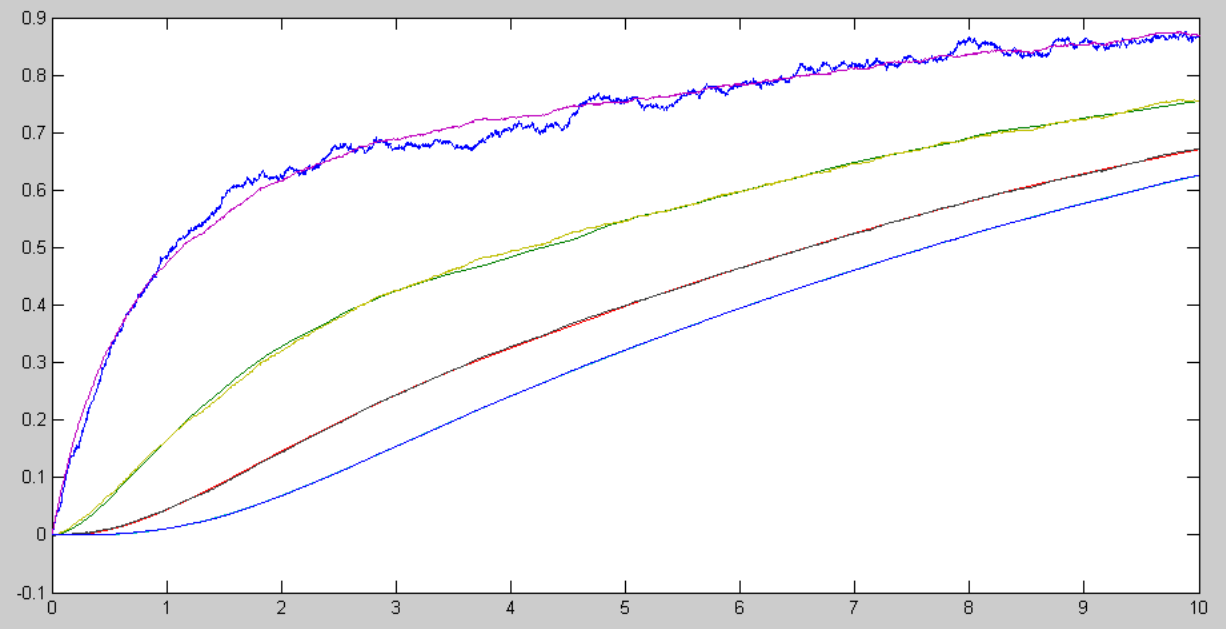
Case 2: Not so Good Sensors (small noise)

- W = 0.1
- V = 1

```
W = 0.01;      % sensor noise
V = 1;         % state disturbance (on U)
Q = F*V*V*F';
R = W*W;
H = lqr(A',  C',  Q,  R)'
A8 = [A, zeros(4,4) ;   H*C,  A - H*C ];
B8 = [B  F  zeros(4,1) ;  B   zeros(4,1)   H];
U = [ ones(N),  randn(N)*V,  randn(N)*W ];
C8 = eye(8,8);
D8 = zeros(8,3);
y = step3(A8, B8, C8, D8, t, X0, U );
plot(t,y)

H =
    1.1169
    1.2204
    0.9184
    0.6843
```

Note that the larger sensor noise has resulted in the observer gains being reduced.



States and State Estimates with State and Sensor Noise: Sensor Noise 10x larger

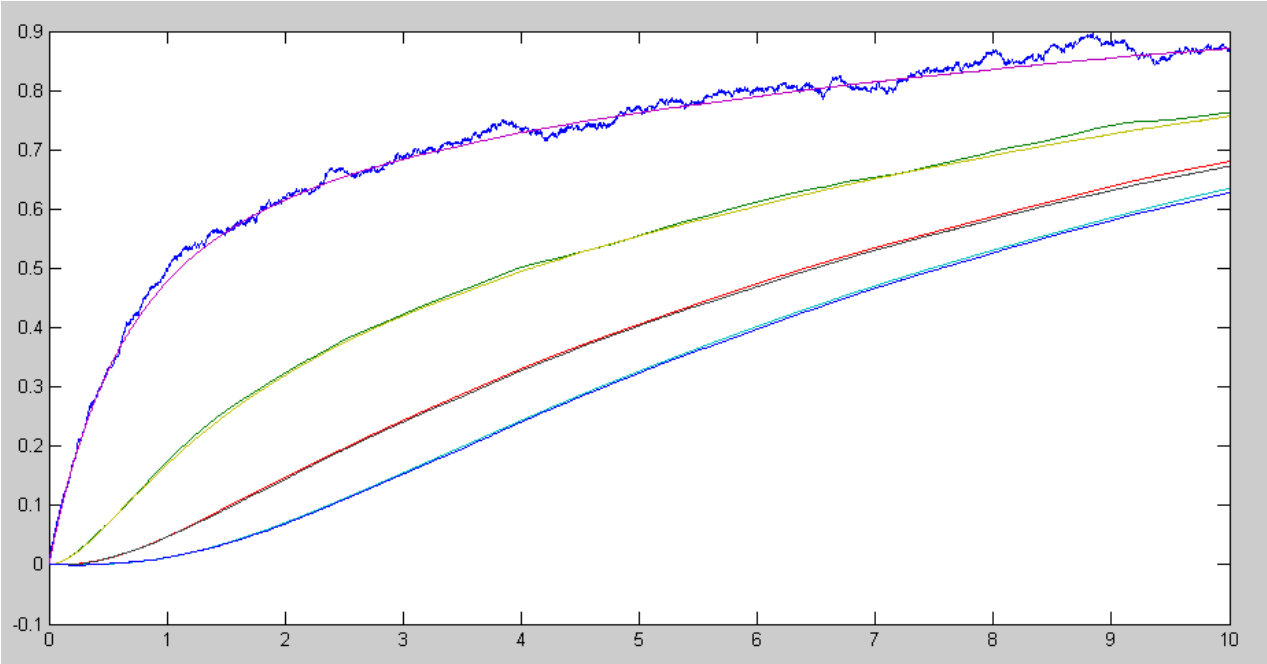
Case 3: Noisy Sensors

- W = 1
- V = 1

```
W = 1;      % sensor noise
V = 1;         % state disturbance (on U)
Q = F*V*V*F';
R = W*W;
H = lqr(A',  C',  Q,  R)'
A8 = [A, zeros(4,4) ;   H*C,  A - H*C ];
B8 = [B  F  zeros(4,1) ;  B   zeros(4,1)   H];
U = [ ones(N),  randn(N)*V,  randn(N)*W ];
C8 = eye(8,8);
D8 = zeros(8,3);
D8(4,3) = 0;
y = step3(A8, B8, C8, D8, t, X0, U );
plot(t,y)

H =

    0.0293
    0.0417
    0.0436
    0.0427
```



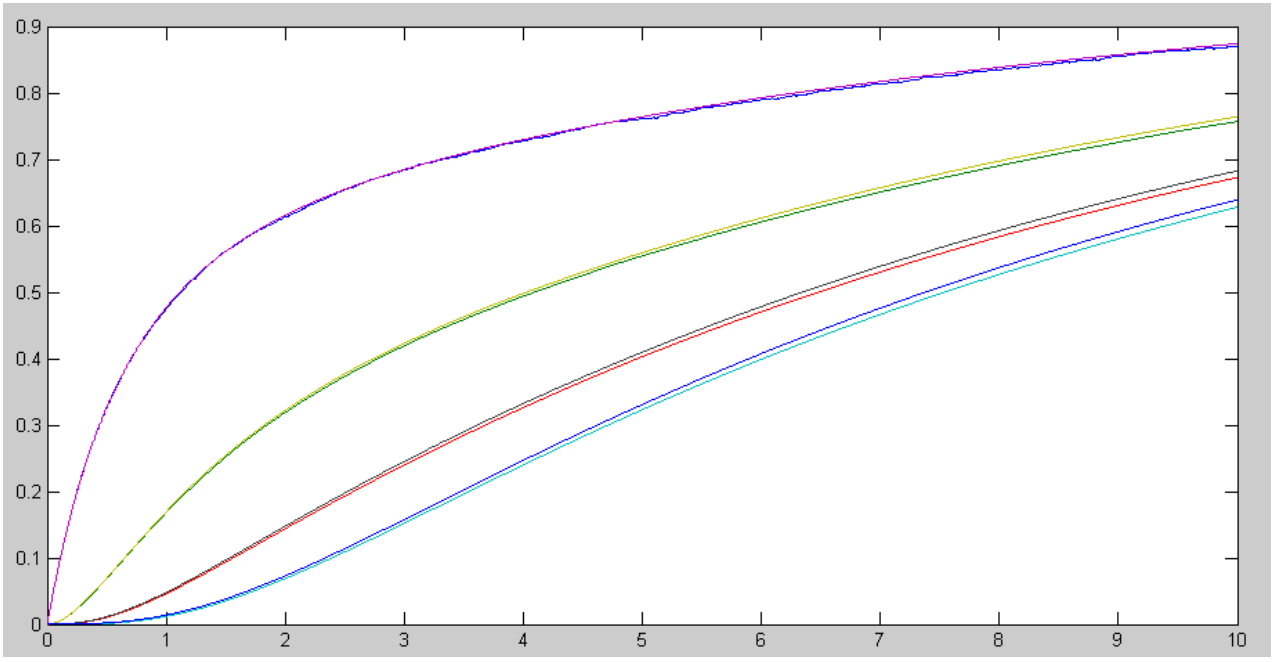
States and State Estimates with State and Sensor Noise: Sensor Noise 100x larger

Note: Only the ratio of Q and R matter. If both disturnances are 10x smaller, you get the same observer gains (same Kalman filter gains)

```
W = 0.1;          % sensor noise
V = 0.1;          % state disturbance (on U)
Q = F*V*V*F';
R = W*W;
H = lqr(A',  C',  Q,  R)'
A8 = [A, zeros(4,4) ;   H*C,  A - H*C ];
B8 = [B  F  zeros(4,1) ;  B   zeros(4,1)   H];
U = [ ones(N),  randn(N)*V,  rand(N)*W ];
C8 = eye(8,8);
D8 = zeros(8,3);
y = step3(A8, B8, C8, D8, t, X0, U );
plot(t,y)

H =

    0.0293
    0.0417
    0.0436
    0.0427
```



States and State Estimates with State and Sensor Noise: Both sensor and input noise scaled down 10x

Main Calling Script in Matlab

```
W = 0.01;      % sensor noise
V = 1;         % state disturbance (on U)
Q = F*V*V*F';
R = W*W;
H = lqr(A',  C',  Q,  R)'
A8 = [A, zeros(4,4) ;   H*C,  A - H*C ];
B8 = [B  F  zeros(4,1) ;  B   zeros(4,1)   H];

t = [0:0.001:10]';
N = size(t);
X0 = zeros(8,1);
U = [ ones(N),  randn(N)*V,  randn(N)*W ];
C8 = eye(8,8);
D8 = zeros(8,3);
y = step3(A8, B8, C8, D8, t, X0, U );
plot(t,y)
```

Step3.m

```
function [ y ] = step3( A, B, C, D, t, X0, U )


T = t(2) - t(1);
[m, n] = size(C);

npt = length(t);

Az = expm(A*T);
Bz = B*T;

X = X0;

y = zeros(npt, m);

y(1,:) = (C*X + D * ( U(1,:)' ) )';

for i=2:npt
    X = Az*X + Bz*( U(i,:)' );
    Y = C*X + D * ( U(i,:)' );

    y(i,:) = Y';

   end

end
```