Q1

06 December 2022 20:02

Question 1:

The file **F16s.mat** gives a linear perturbation model for an F16 aircraft in level flight at 10,000 ft. The outputs of this model are the aircraft's speed and climb rate, both in ft/sec. The two inputs to the model are stochastic processes, $d_1(t)$ and $d_2(t)$ representing wind gust effects on the vertical component of the aircraft velocity, and on the pitch rate of the aircraft, respectively.

Using the Dryden model for vertical wind turbulence, we can model $d_1(t)$ and $d_2(t)$ as zero mean WSS processes with spectra

$$S_{d_1d_1}(\omega) = \frac{K(1+3(a\omega)^2)}{(1+(a\omega)^2)^2}$$

and

$$S_{d_2d_2}(\omega) = \left[\frac{\omega^2}{1+(b\omega)^2}\right] S_{d_1d_1}(\omega)$$

Moreover, these disturbances are assumed to be correlated, which can be modeled by using a single white noise source to drive the shaping filters for both disturbances. (Hence d_2 can be modeled by passing d_1 through an additional shaping filter, given the structure of its spectrum.) For the flight conditions above, the values $a = 3$, $b = 0.1$, are appropriate.

a.) The constant K controls the RMS (root-mean-square) magnitude of the vertical gust component. Choose this constant so that $\sqrt{E[d_1(t)^2]} = 7$.

b.) With the value of K found in a.), compute the RMS magnitude of the variations in speed and climb rate that result from the disturbance inputs.

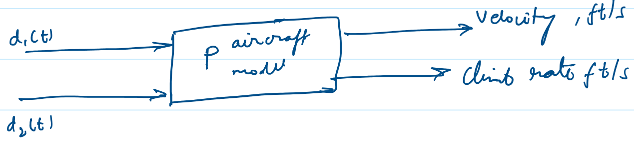
c.) Compute the components F and Q for the stochastic discretization of the complete system. Assume a sample rate of 20Hz.

d.) Show that the discretization computed in c.) is consistent with the calculations in b.). That is, show that the discrete output sequence $\mathbf{y}_k$ has the same steady-state variance as found in b.)

a)

Given

h = 1000 ft



zero mean

WSS

$$S_{d_1d_1}(\omega) = \frac{K(1+3(a\omega)^2)}{(1+(a\omega)^2)^2}$$

where $a = 3$, $\omega^2 = -s^2$

$$\Rightarrow S_{dd}(s) = \frac{K(1+3a^2(-s^2))}{(1-9s^2)^2}$$

$$= \frac{K(1-3a^2s^2)}{(1-9s^2)^2}$$

$$= \frac{K(1-27s^2)}{(1-9s^2)^2}$$

$\because$ WSS,

$$R(\tau) = \mathcal{L}^{-1}\{S_{d_1d_1}(s)\}$$

```
syms s K
f = K*(1 - 27*s^2)/((1 - 9*s^2)^2)
R_tau = ilaplace(f)
```

f =

$$-\frac{K\,(27\,s^2-1)}{(9\,s^2-1)^2}$$

R_tau =

$$\frac{K\,e^{-\frac{t}{3}}}{3} - \frac{K\,e^{t/3}}{3} - \frac{K\,t\,e^{-\frac{t}{3}}}{18} - \frac{K\,t\,e^{t/3}}{18}$$

$$= k\left\{\frac{e^{-\tau/3}}{3} - \frac{\tau e^{-\tau/3}}{18}\right\} \mathbb{1}(\tau)$$

$$+ k\left\{\frac{e^{-\tau/3}}{3} - \frac{\tau e^{-\tau/3}}{18}\right\} \mathbb{1}(-\tau)$$

$$R(\tau) = \begin{cases} k\left(\frac{e^{-\tau/3}}{3} - \frac{\tau e^{-\tau/3}}{18}\right), & \tau > 0 \\ k\left(\frac{e^{\tau/3}}{3} - \frac{\tau e^{\tau/3}}{18}\right), & \tau < 0 \end{cases}$$

$$\boxed{\therefore \quad E[d_1(t)^2] = R(0) = k/3} \quad (1)$$

Given,

$$\sqrt{R(0)} = 7, \quad \therefore R(0) = 49$$

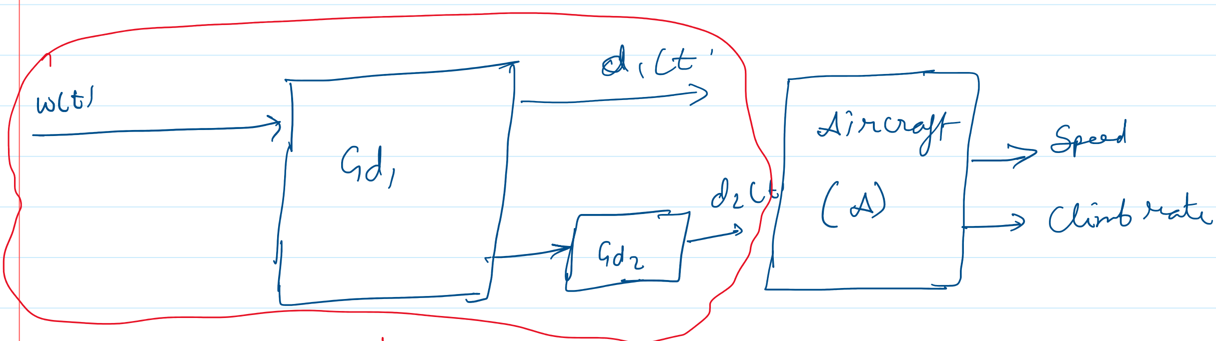
putting above,

$$k/3 = 49$$

$$\boxed{\therefore \quad k = 147}$$

b)

We know that the augmented model for observer will be given by,



Assuming Z_1 & Z_2 to be the state vectors we get,

$$\dot{z}_1 = A d_1 Z_1 + B d_1 w, \quad Y_{d_1} = C d_1 Z$$

$\dot{z}_1 = \ldots, \quad Y_{d_1} = C_{d_1} z$

$\dot{z}_2 = A_{d_2} z_2 + B_{d_2} w, \quad Y_{d_2} = C_{d_2} z$

Here, $A_{d_1}, B_{d_1}, C_{d_1}$ for d_1 output comes from G_{d_1},

$$S_{d_1 d_1} = \frac{3\sqrt{3}s+1}{(3s+1)^2} \times 147 \times \frac{(1-3\sqrt{3}s)}{(1-3s)^2}$$

stable part

taking $Q = 147$ for S_{ww} ($\because w(t)$ is white)

```
bd1 = [0 3*sqrt(3) 1]
ad1 = [9 6 1]

[Ad1, Bd1, Cd1, Dd1] = tf2ss(bd1,ad1)
```

```
bd2 = [0 5.1962 1 0]
ad2 = [0.9 9.6 6.1 1]

[Ad2, Bd2, Cd2, Dd2] = tf2ss(bd2,ad2)
```

Augmented model for disturbance

```
bd1 = 1×3
        0    5.1962    1.0000

ad1 = 1×3
     9     6     1

Ad1 = 2×2
   -0.6667   -0.1111
    1.0000         0

Bd1 = 2×1
     1
     0

Cd1 = 1×2
    0.5774    0.1111
```

and $A_{d_2}, B_{d_2}, C_{d_2}$ for d_2 output comes from $G_{d_2} \times G_{d_1}$,

$$S_{d_2 d_2}(s) = \left[\frac{-s^2}{1-b^2 s^2}\right] S_{d_1 d_1}(s), \quad b = 0.1$$

$$\left(\frac{s}{1+bs}\right)\left(\frac{-s}{1-bs}\right)$$

$$S_{d_2 d_2}(s) = \frac{(3\sqrt{3}s^2 + s)}{\underbrace{(0.9s^3 + 9.6s^2 + 6.1s + 1)}_{\text{stable part}}} \times \underbrace{147}_{Q} \times \frac{(3\sqrt{3}s^2 - s)}{(-0.9s^3 + 9.6s^2 - 6.1s + 1)}$$

$\because w(t)$ is white

```
bd2 = [0 5.1962 1 0]
ad2 = [0.9 9.6 6.1 1]

[Ad2, Bd2, Cd2, Dd2] = tf2ss(bd2,ad2)
```

```
bd2 = 1×4
        0    5.1962    1.0000 ...

ad2 = 1×4
    0.9000    9.6000    6.1000 ...

Ad2 = 3×3
  -10.6667   -6.7778   -1.1111
    1.0000         0         0
         0    1.0000         0

Bd2 = 3×1
     1
     0
     0

Cd2 = 1×3
```

```
bd2 = [0 5.1962 1 0]
ad2 = [0.9 9.6 6.1 1]

[Ad2, Bd2, Cd2, Dd2] = tf2ss(bd2,ad2)
```

```
bd2 = 1×4
         0    5.1962    1.0000 ...

ad2 = 1×4
    0.9000    9.6000    6.1000 ...

Ad2 = 3×3
  -10.6667   -6.7778   -1.1111
    1.0000         0         0
         0    1.0000         0

Bd2 = 3×1
     1
     0
     0

Cd2 = 1×3
    5.7736    1.1111         0

Dd2 = 0
```

: w(t) is white

Stacking $\dot{z}_1$, $\dot{z}_2$ into z, we get,

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} = \underbrace{\begin{bmatrix} Ad_1 & 0 \\ 0 & Ad_2 \end{bmatrix}}_{Ad} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + \underbrace{\begin{bmatrix} Bd_1 \\ Bd_2 \end{bmatrix}}_{Bd} w$$

$$\overbrace{\begin{bmatrix} d_1 \\ d_2 \end{bmatrix}}^{Y} = \underbrace{\begin{bmatrix} Cd_1 & 0 \\ 0 & Cd_2 \end{bmatrix}}_{Cd} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$$

Augmented model for disturbance

```
Ad = [Ad1 zeros(2,3); zeros(3,2) Ad2]
Bd = [Bd1;Bd2]
Cd = [Cd1 zeros(1,3); zeros(1,2) Cd2]
Dd = [0;0]
```

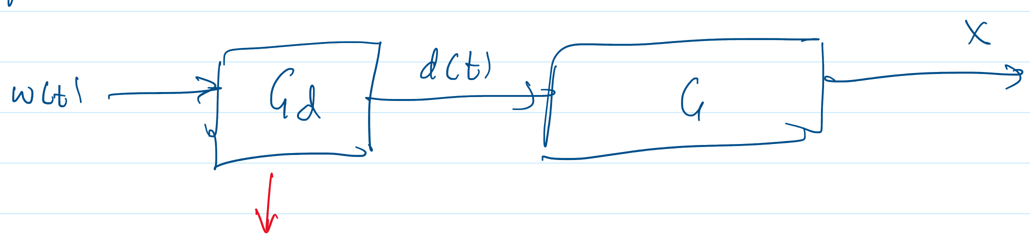
```
Ad = 5×5
   -0.6667   -0.1111         0         0         0
    1.0000         0         0         0         0
         0         0  -10.6667   -6.7778   -1.1111
         0         0    1.0000         0         0
         0         0         0    1.0000         0

Bd = 5×1
     1
     0
     1
     0
     0

Cd = 2×5
    0.5774    0.1111         0         0         0
         0         0    5.7736    1.1111         0

Dd = 2×1
     0
     0
```

So, now basically we can write in simple form as,



corresponds to A_d found above.

$\therefore$ Augmented System will be given by,

$$\frac{d}{dt}\begin{bmatrix} \underline{X} \\ \underline{Z} \end{bmatrix} = \underbrace{\begin{bmatrix} A & BC_d \\ 0 & A_d \end{bmatrix}}_{A_a}\begin{bmatrix} \underline{X} \\ \underline{Z} \end{bmatrix} + \overbrace{\begin{bmatrix} BD_d \\ B_d \end{bmatrix}}^{B_a}\underline{w}$$

$$Y = \underbrace{\begin{bmatrix} C & 0 \end{bmatrix}}_{C_a}\begin{bmatrix} \underline{X} \\ \underline{Z} \end{bmatrix}$$

Augmented oberver matrix for

```
Aa = [A B*Cd; zeros(5,4) Ad]
Ba = [B*Dd; Bd]
Ca = [C zeros(2,5)]
```

Aa = 9×9

-0.0083	-0.0703	-0.0581	0.0277	0.0344	0.0066	-0.1026	-0.0198	0
0.0770	-0.0082	-0.0750	0.0345	-0.0293	-0.0056	0.1669	0.0321	0
-0.0612	0.0714	-1.5824	2.0070	0.1731	0.0333	0.5542	0.1067	0
0.0150	-0.0282	-1.1581	-0.7484	-0.0129	-0.0025	0.8395	0.1616	0
0	0	0	0	-0.6667	-0.1111	0	0	0
0	0	0	0	1.0000	0	0	0	0
0	0	0	0	0	0	-10.6667	-6.7778	-1.1111
0	0	0	0	0	0	1.0000	0	0
0	0	0	0	0	0	0	1.0000	0

Ba = 9×1

0
0
0
0
1
0
1
0
0

Ca = 2×9

0.0365	-0.0415	-0.0087	0.0175	0	0	0	0	0
0.0504	0.0410	0.3147	-0.1461	0	0	0	0	0

Now, to find the variance by applying Stochastic Disc., and Kalman Filter,

$$Q\_c = B_a \times \overset{147}{Q} \times B_a^T$$

0

Q_c = 9×9

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	147	0	147	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	147	0	147	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

Variance of the process using lyapunov equation

```
Q_c = Ba*147*Ba'
Pss = lyap(Aa, Q_c)

%%RMS values
Var = Ca*Pss*Ca'
rms_speed = sqrt(Var(1,1))
rms_climbrate = sqrt(Var(2,2))
```

Pss = 9×9

30.7618	0.0234	-0.3238	0.2390	-9.2926	76.0061	-0.2802	-0.9012	7.6907
0.0234	27.1250	0.1807	-0.2737	1.6062	-38.5397	0.4234	0.1183	-3.8658
-0.3238	0.1807	1.1326	0.0337	9.1116	0.0073	0.0310	0.9081	-0.0901
0.2390	-0.2737	0.0337	0.9012	-2.1216	-14.5201	0.8174	-0.2939	-1.4226
-9.2926	1.6062	9.1116	-2.1216	110.2500	0.0000	6.9982	10.3252	-1.0325
76.0061	-38.5397	0.0073	-14.5201	0.0000	992.2500	-10.3252	1.0325	99.1217
-0.2802	0.4234	0.0310	0.8174	6.9982	-10.3252	6.9982	-0.0000	-1.0325
-0.9012	0.1183	0.9081	-0.2939	10.3252	1.0325	-0.0000	1.0325	0.0000
7.6907	-3.8658	-0.0901	-1.4226	-1.0325	99.1217	-1.0325	0.0000	9.9122

Var = 2×2

0.0891	-0.0037
-0.0037	0.2463

$\boxed{\text{rms\_speed = 0.2984} \\ \text{rms\_climbrate = 0.4963}}$

```
0.0891   -0.0037
-0.0037   0.2463

rms_speed = 0.2984
rms_climbrate = 0.4963
```

∴ RMS for variation in speed,

$$= 0.2984$$

RMS for variation in climb rate

$$= 0.4963$$

c)

Given, Sample rate $= 20\,Hz$

$$\therefore \tau = 1/20\ s = 0.05\ s$$

using van-loan's algorithm,

$$A = \left[\begin{array}{c|c} -A_a & B_a Q B_a^T \\ \hline 0 & A^T \end{array}\right]$$

$$e^{A\tau} = \left[\begin{array}{c|c} \angle \text{mess} & F^{-1}\bar{Q} \\ \hline 0 & F^T \end{array}\right]$$

Variance of the process using lyapunov equation

```
Q_c = Ba*147*Ba';
Pss = lyap(Aa, Q_c);

%%RMS values
Var = Ca*Pss*Ca';
rms_speed = sqrt(Var(1,1));
rms_climbrate = sqrt(Var(2,2));
```

```
%%c part
lam = [-Aa Q_c; zeros(9,9) Aa']
Tau = 0.05
stat_trans = expm(lam*Tau)
F = stat_trans(10:18,10:18)'
Q_bar = F*stat_trans(1:9,10:18)
```

```
  :
Tau = 0.0500
stat_trans = 18×18
    1.0004    0.0035    0.0030   -0.0016   -0.0018   -0.0003    0.0067    0.0020    0.0002 ..
   -0.0038    1.0004    0.0038   -0.0019    0.0015    0.0003   -0.0110   -0.0033   -0.0003
    0.0032   -0.0038    1.0792   -0.1063   -0.0091   -0.0018   -0.0350   -0.0106   -0.0009
   -0.0007    0.0013    0.0613    1.0351    0.0004    0.0001   -0.0570   -0.0171   -0.0014
         0         0         0         0    1.0338    0.0056         0         0         0
         0         0         0         0   -0.0508    0.9999         0         0         0
         0         0         0         0         0         0    1.6925    0.4448    0.0732
         0         0         0         0         0         0   -0.0659    0.9898   -0.0017
         0         0         0         0         0         0    0.0015   -0.0498    1.0000
         0         0         0         0         0         0         0         0         0
  :
```

→ F

```
F = 9×9
    0.9996   -0.0035   -0.0028    0.0012    0.0017    0.0003   -0.0040   -0.0003    0.0001
    0.0039    0.9996   -0.0037    0.0015   -0.0015   -0.0003    0.0065    0.0004   -0.0002
   -0.0029    0.0034    0.9212    0.0946    0.0082    0.0016    0.0223    0.0014   -0.0007
    0.0008   -0.0015   -0.0546    0.9605   -0.0009   -0.0002    0.0313    0.0019   -0.0010
         0         0         0         0    0.9671   -0.0055         0         0         0
         0         0         0         0    0.0492    0.9999         0         0         0
         0         0         0         0         0         0    0.5807   -0.2631   -0.0429
         0         0         0         0         0         0    0.0386    0.9928   -0.0012
         0         0         0         0         0         0    0.0011    0.0499    1.0000
```

→ Q

```
Q_bar = 9×9
    0.0000   -0.0000   -0.0002   -0.0002   -0.0095   -0.0003   -0.0069   -0.0003   -0.0000
   -0.0000    0.0001    0.0004    0.0004    0.0201    0.0007    0.0146    0.0005    0.0000
   -0.0002    0.0004    0.0025    0.0026    0.1162    0.0039    0.0840    0.0032    0.0001
   -0.0002    0.0004    0.0026    0.0027    0.1214    0.0040    0.0881    0.0033    0.0001
   -0.0095    0.0201    0.1162    0.1214    7.1097    0.1777    5.5950    0.1515    0.0026
   -0.0003    0.0007    0.0039    0.0040    0.1777    0.0060    0.1279    0.0050    0.0001
   -0.0069    0.0146    0.0840    0.0881    5.5950    0.1279    4.4975    0.1098    0.0018
   -0.0003    0.0005    0.0032    0.0033    0.1515    0.0050    0.1098    0.0042    0.0001
   -0.0000    0.0000    0.0001    0.0001    0.0026    0.0001    0.0018    0.0001    0.0000
```

d) To apply dlyap we need to check the magnitude of eigen values first,

$$\underline{x_{k+1}} = F_k \underline{x_k} + \underline{w_k}$$

$$E[\underline{w_j} w_k^T] = Q_k \delta_{jk}$$

$\Rightarrow |\lambda| < 1$

d part

```
abs(eig(F))
```

```
ans = 9×1
    0.9996
    0.9996
    0.9434
    0.9434
    0.9835
    0.9835
    0.6065
    0.9835
    0.9835
```

∴ using "dlyap" in matlab,

```
G = eye(9,9)
Q_d = G*Q_bar*G'
```

```
G = 9×9
     1     0     0     0     0     0     0     0     0
     0     1     0     0     0     0     0     0     0
     0     0     1     0     0     0     0     0     0
     0     0     0     1     0     0     0     0     0
     0     0     0     0     1     0     0     0     0
     0     0     0     0     0     1     0     0     0
     0     0     0     0     0     0     1     0     0
     0     0     0     0     0     0     0     1     0
     0     0     0     0     0     0     0     0     1

Q_d = 9×9
    0.0000   -0.0000   -0.0002   -0.0002   -0.0095   -0.0003   -0.0069   -0.0003   -0.0000
   -0.0000    0.0001    0.0004    0.0004    0.0201    0.0007    0.0146    0.0005    0.0000
   -0.0002    0.0004    0.0025    0.0026    0.1162    0.0039    0.0840    0.0032    0.0001
   -0.0002    0.0004    0.0026    0.0027    0.1214    0.0040    0.0881    0.0033    0.0001
   -0.0095    0.0201    0.1162    0.1214    7.1097    0.1777    5.5950    0.1515    0.0026
   -0.0003    0.0007    0.0039    0.0040    0.1777    0.0060    0.1279    0.0050    0.0001
   -0.0069    0.0146    0.0840    0.0881    5.5950    0.1279    4.4975    0.1098    0.0018
   -0.0003    0.0005    0.0032    0.0033    0.1515    0.0050    0.1098    0.0042    0.0001
   -0.0000    0.0000    0.0001    0.0001    0.0026    0.0001    0.0018    0.0001    0.0000
```

```
G = eye(9,9)
Q_d = G*Q_bar*G'
P_disc = dlyap(F,Q_d)
var_discrete = Ca*P_disc*Ca'
rms_speed = sqrt(var_discrete(1,1))
rms_pitchrate = sqrt(var_discrete(2,2))
```

R

```
G = 9×9
     1     0     0     0     0     0     0     0     0
     0     1     0     0     0     0     0     0     0
     0     0     1     0     0     0     0     0     0
     0     0     0     1     0     0     0     0     0
     0     0     0     0     1     0     0     0     0
     0     0     0     0     0     1     0     0     0
     0     0     0     0     0     0     1     0     0
     0     0     0     0     0     0     0     1     0
     0     0     0     0     0     0     0     0     1

Q_d = 9×9
    0.0000   -0.0000   -0.0002   -0.0002   -0.0095   -0.0003   -0.0069   -0.0003   -0.0000
   -0.0000    0.0001    0.0004    0.0004    0.0201    0.0007    0.0146    0.0005    0.0000
   -0.0002    0.0004    0.0025    0.0026    0.1162    0.0039    0.0840    0.0032    0.0001
   -0.0002    0.0004    0.0026    0.0027    0.1214    0.0040    0.0881    0.0033    0.0001
   -0.0095    0.0201    0.1162    0.1214    7.1097    0.1777    5.5950    0.1515    0.0026
   -0.0003    0.0007    0.0039    0.0040    0.1777    0.0060    0.1279    0.0050    0.0001
   -0.0069    0.0146    0.0840    0.0881    5.5950    0.1279    4.4975    0.1098    0.0018
   -0.0003    0.0005    0.0032    0.0033    0.1515    0.0050    0.1098    0.0042    0.0001
   -0.0000    0.0000    0.0001    0.0001    0.0026    0.0001    0.0018    0.0001    0.0000

P_disc = 9×9
   30.7618    0.0234   -0.3238    0.2390   -9.2926   76.0061   -0.2802   -0.9012    7.6907
    0.0234   27.1250    0.1807   -0.2737    1.6062  -38.5397    0.4234    0.1183   -3.8658
   -0.3238    0.1807    1.1326    0.0337    9.1116    0.0073    0.0310    0.9081   -0.0901
    0.2390   -0.2737    0.0337    0.9012   -2.1216  -14.5201    0.8174   -0.2939   -1.4226
   -9.2926    1.6062    9.1116   -2.1216  110.2500    0.0000    6.9982   10.3252   -1.0325
   76.0061  -38.5397    0.0073  -14.5201    0.0000  992.2500  -10.3252    1.0325   99.1217
   -0.2802    0.4234    0.0310    0.8174    6.9982  -10.3252    6.9982    0.0000   -1.0325
   -0.9012    0.1183    0.9081   -0.2939   10.3252    1.0325    0.0000    1.0325   -0.0000
    7.6907   -3.8658   -0.0901   -1.4226   -1.0325   99.1217   -1.0325   -0.0000    9.9122

var_discrete = 2×2
    0.0891   -0.0037
   -0.0037    0.2463

rms_speed = 0.2984
rms_pitchrate = 0.4963
```

MS variation in speed = 0.2984

Rms variation in climb rate = 0.4963

The values are consistent to what we got in 'b' part.

Q2

06 December 2022 15:40

Question 2:

Suppose in Question #1 that a pitot tube can measure the speed of the aircraft at a rate of 20Hz. These measurements are corrupted by noise, modeled as white and WSS with a variance of 8 ft/sec. The measurements are uncorrelated with the disturbances at all times.

Assume that a Kalman filter is implemented to estimate the states based on these measurements. Letting $\mathbf{x}(t)$ be the states of the complete physical model, assume that $E[\mathbf{x}(0)] = 0$ and $E[\mathbf{x}(0)\mathbf{x}(0)^T] = \frac{1}{2}\boldsymbol{I}$. Plot the evolution in RMS uncertainty of the true values of the two outputs (i.e. $E[\tilde{y}_1(t)^2]$ and $E[\tilde{y}_2(t)^2]$). What is the steady-state RMS uncertainty in each output?

HINT: Note that the measurements (airspeed) used by the filter in this problem are different from the outputs we are interested in estimating! This poses no conceptual difficulties; just use a $\boldsymbol{C}$ matrix in the design of the estimator that reflects the actual measurement, then use the original $\boldsymbol{C}$ matrix to assess the output estimation error variance.

Given:- $\quad y_k = H_k x_k + v$

$$R_k = 8 \text{ ft/sc} = \underset{\sim}{var}[v]$$

$\because$ only one measurement

$$E[y_k v_k] = 0$$

$$\underset{\sim}{x}(t) = \begin{bmatrix} 9\times 1 \end{bmatrix}, \text{ from the augmented model in Q1,}$$

$$E[\underset{\sim}{x}(0)] = 0, \quad P_0 = E[\underset{\sim}{x}(0)\underset{\sim}{x}(0)^T] = \frac{1}{2}\mathbb{I}$$

$$\Downarrow$$

$$\hat{\underset{\sim}{x}}_0^- = 0$$

$$P_k = E[\tilde{\underset{\sim}{x}}_k \tilde{\underset{\sim}{x}}_k^T], \quad \text{where, } \tilde{\underset{\sim}{x}}_k = \underset{\sim}{x}_k - \hat{\underset{\sim}{x}}_k$$

To find,

$$E[\tilde{y}_1(t)^2], \quad E[\tilde{y}_2(t)^2],$$

$$\Rightarrow \quad C_a P_k C_a'$$

augmented C_a from Q1

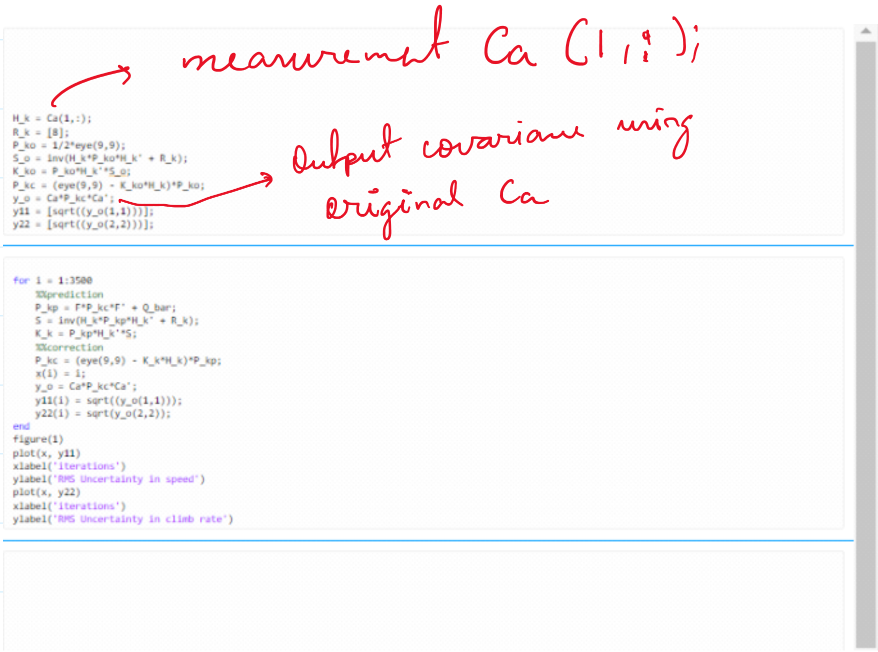
using the Kalman Filter algorithm,

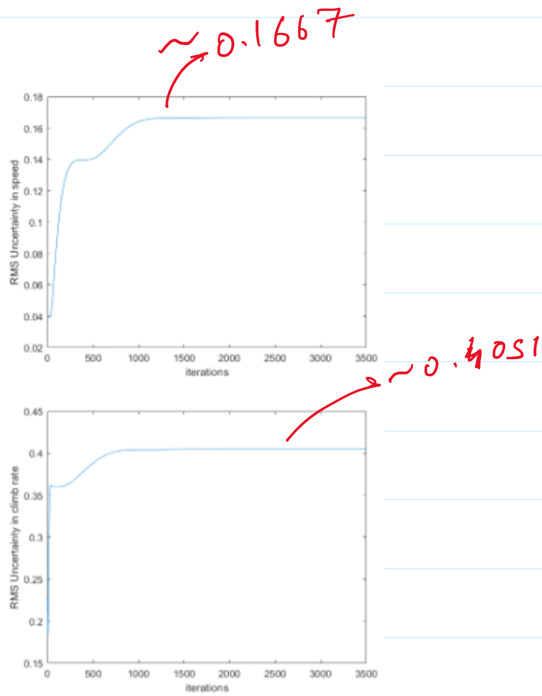
H_k = 1st row of Ca of augmented matrix. (∵ one measurement available)

Coding the algorithm in matlab

```
H_k = Ca(1,:);
R_k = [8];
P_ko = 1/2*eye(9,9);
S_o = inv(H_k*P_ko*H_k' + R_k);
K_ko = P_ko*H_k'*S_o;
P_kc = (eye(9,9) - K_ko*H_k)*P_ko;
y_o = Ca*P_kc*Ca';
y11 = [sqrt((y_o(1,1)))];
y22 = [sqrt((y_o(2,2)))];
```

```
for i = 1:3500
    %%prediction
    P_kp = F*P_kc*F' + Q_bar;
    S = inv(H_k*P_kp*H_k' + R_k);
    K_k = P_kp*H_k'*S;
    %%correction
    P_kc = (eye(9,9) - K_k*H_k)*P_kp;
    x(i) = i;
    y_o = Ca*P_kc*Ca';
    y11(i) = sqrt((y_o(1,1)));
    y22(i) = sqrt(y_o(2,2));
end
figure(1)
plot(x, y11)
xlabel('iterations')
ylabel('RMS Uncertainty in speed')
plot(x, y22)
xlabel('iterations')
ylabel('RMS Uncertainty in climb rate')
```





using "dare" function to check the steady state using "dare" function to check the steady state RMS uncertainty in each output,

$$P_\infty^- = \text{dare}(F', H', Q, R)$$

using Discrete algebraic Ricatti equation.

$$K_\infty = P_\infty^- \times H_k \times (H_k \times P_\infty^- \times H_k^T + R_k)$$

$$P_\infty = (\mathbb{I} - K_\infty H_k) P_\infty^-$$

```
%%%dare to
[Ppinf,L_,G_] = dare(F',H_k',Q_bar,R_k)
Sinf = inv(H_k*Ppinf*H_k' + R_k);
K_inf = Ppinf*H_k'*S
P_cinf = (eye(9,9) - K_inf*H_k)*Ppinf;
var = Ca*P_cinf*Ca'
```

```
iterations

Ppinf = 9×9
    9.8095   -0.0580   -0.2206   -0.3488   -5.9368   59.3944 ...
   -0.0580    8.7122    0.1480    0.3525    0.1298  -29.9727
   -0.2206    0.1480    1.1309    0.0365    9.0615    0.3169
   -0.3488    0.3525    0.0365    0.8616   -1.9959  -15.1075
   -5.9368    0.1298    9.0615   -1.9959  107.9913   12.2599
   59.3944  -29.9727    0.3169  -15.1075   12.2599  922.2464
   -0.3180    0.4250    0.0308    0.8156    7.0108  -10.3520
   -0.5619   -0.0295    0.9031   -0.2811   10.0981    2.2612
    5.9956   -2.9943   -0.0586   -1.4826    0.2162   91.9985

L_ = 9×1 complex
   0.9979 + 0.0030i
   0.9979 - 0.0030i
   0.9835 + 0.0000i
   0.9835 + 0.0000i
   0.9834 + 0.0028i
   0.9834 - 0.0028i
   0.9409 + 0.0691i
   0.9409 - 0.0691i
   0.6065 + 0.0000i

G_ = 1×9
    0.0446   -0.0446   -0.0029   -0.0014   -0.0427    0.3897 ...

K_inf = 9×1
    0.0444
   -0.0447
   -0.0029
   -0.0016
   -0.0419
    0.3918
   -0.0019
   -0.0040
    0.0396

var = 2×2
    0.0278   -0.0023
   -0.0023    0.1641
```

$$\boxed{\therefore \; E[\tilde{y}_1(t)^2] = \sqrt{0.0278} \simeq 0.16673}$$

(speed at steady state)

$$\boxed{E[\tilde{y}_2(t)^2] = \sqrt{0.1641} \simeq 0.40509}$$

(climb rate at steady state)

Q3

06 December 2022 22:18

Question 3:

On landing approach, proper knowledge (and control) of the descent rate is crucial. Suppose that the RMS uncertainty in the climb rate must below ±0.5 ft/sec to ensure safe landing. Can this be assured with the given measurements? If not, how small would the variance in each speed measurement need to be to ensure this level of uncertainty in the climb rate? Conversely, if the given measurements can assure this level, how "bad" could the pitot tube measurements to be (i.e. how large can the variance of the measurement noise be) and still provide the required RMS uncertainty level?

Note: The aircraft speed, and the disturbance spectra, would generally be different at landing as opposed to the cruise condition considered above. We'll ignore these effects, rather than recalculate everything for a new flight condition!

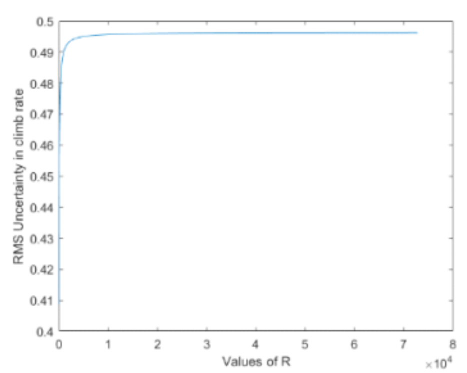
The RMS uncertainty in climb rate is below ±0.5 ft/s, hence safe landing can be ensured with the given measurements, with R = 8 ft/sec for noise.

Varying R to see the output in climb rate,

```
%%%Question3

for j = 1:50
    R_k = R_k*1.2
    [Ppinf,L_,G_] = dare(F',H_k',Q_bar,R_k)
    Sinf = inv(H_k*Ppinf*H_k' + R_k);
    K_inf = Ppinf*H_k'*S
    P_cinf = (eye(9,9) - K_inf*H_k)*Ppinf;
    var = Ca*P_cinf*Ca'
    x_R(j) = R_k
    y(j) = sqrt(var(2,2))

end
figure(1)
plot(x_R,y)
xlabel('Values of R')
ylabel('RMS Uncertainty in climb rate')
```



It can be seen that ↑ing R_k doesn't effect the uncertainty in climb rate.

```
R_k = 1000000;
[Ppinf,L_,G_] = dare(F',H_k',Q_bar,R_k);
Sinf = inv(H_k*Ppinf*H_k' + R_k);
K_inf = Ppinf*H_k'*S;
P_cinf = (eye(9,9) - K_inf*H_k)*Ppinf;
var = Ca*P_cinf*Ca';
sqrt(var(2,2))
```

```
ans = 0.4963
```

Value of RMS uncertainty in climb rate even when variance in measurement noise is 1000000

Hence, even when the variance is infinite the required RMS uncertaintly level is attained.

The reason is that system is able to tolerate any amount of noise, since the climb rate RMS uncertainty never crosses 0.5.

" An infinite amount of noise will render the measurements helpful and rely only on propogation." (open loop).

Basically Kalman gain goes to zero. as $R_k \to \infty$

$$K_k = \underbrace{P_k^- H_k^T \left[H_k P_k^- H_k^T + R_k \right]^{-1}}_{= 0 \text{ when } R_k = \infty}$$

$\downarrow \infty$

$$\boxed{\text{Hence, } \quad \hat{\underline{x}}_k = \hat{\underline{x}}_k^- + 0 \left[\underline{y}_k - H_k \hat{\underline{x}}_k^- \right]}$$

$$\Rightarrow \boxed{\hat{\underline{x}}_k = \hat{\underline{x}}_k^-}$$