Simulation results on Kalman filter, Paper B15

Yang ZHENG

Q4.

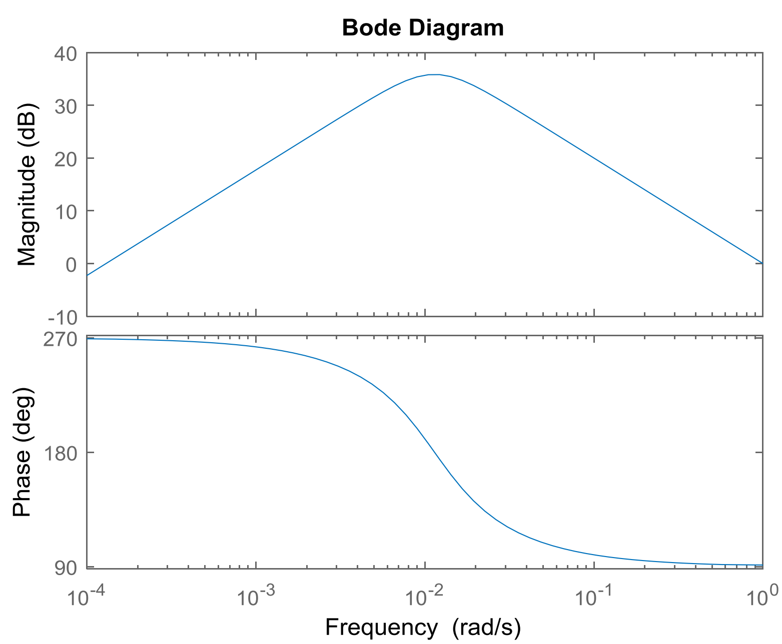
```
%% Q4
clc;clear; close all
m = 7700;

A = [0 -1/m; 0 0]; F = [0;1]; V = 1; W = 1; C = [1 0];

% X = are(A, B, C) returns the stablizing solution (if it
%     exists) to the continuous-time Riccati equation:
%           A'*X + X*A - X*B*X + C = 0

P = are(A',C'*W^(-1)*C,F*V*F')
Kf = P*C'*W^(-1)

v0 = 40;
%% b)
A1 = [0 -1/m; 0 -0.6*v0^2/m];
P1 = are(A1',C'*W^(-1)*C,F*V*F')
Kf1 = P1*C'*W^(-1)
```

Filter frequency response:

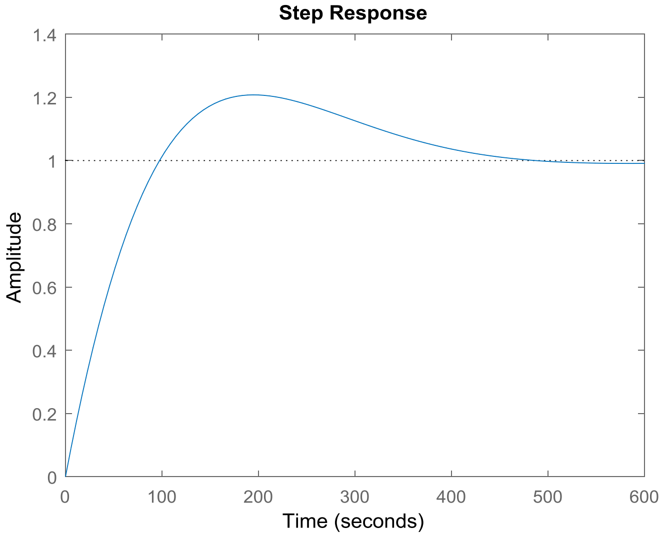
The first Kalman filter: constant aero-drag model

$$K_{f1} = \begin{bmatrix} 0.0161 \\ -1 \end{bmatrix}$$

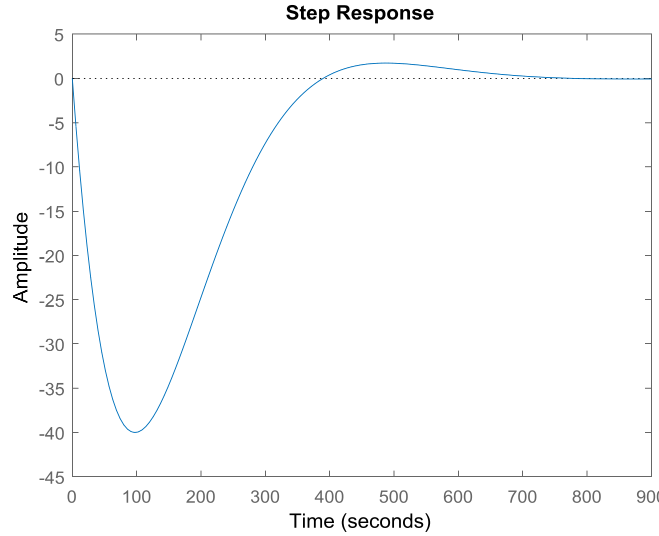
Transfer function from the velocity measurement y to the estimate of aero-drag D_a



Step response (unite change of velocity measurement; check the output of aero-drag)



(a) estimate of velocity

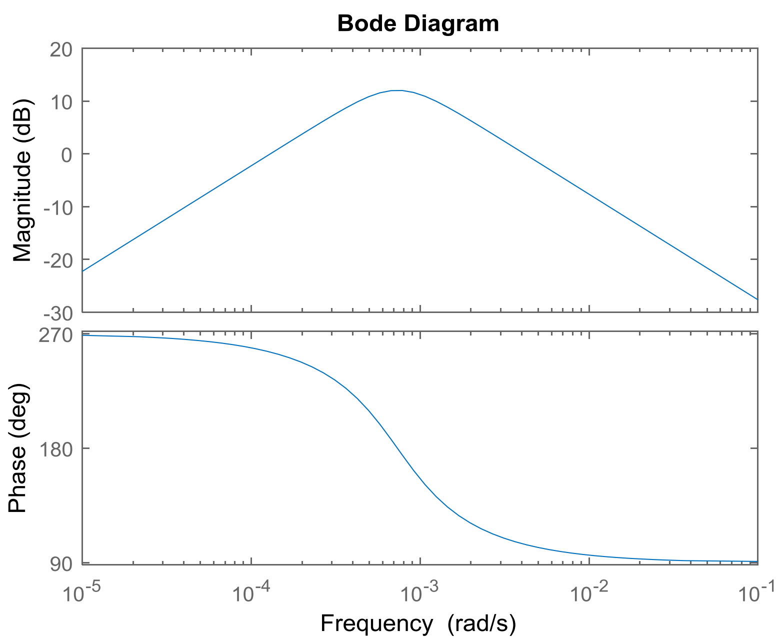


(b) estimate of aero-drag

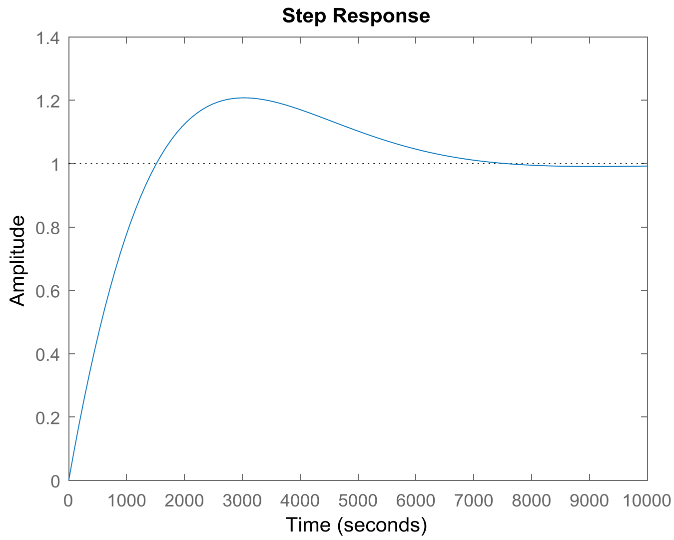
The second Kalman filter: non-constant aero-drag model

$$K_{f1} = \begin{bmatrix} 0.001 \\ -0.0041 \end{bmatrix}$$

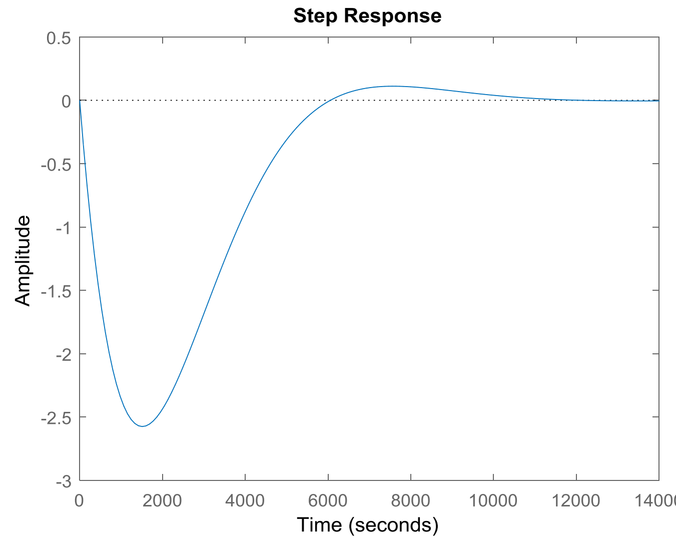
Transfer function from the velocity measurement y to the estimate of aero-drag D_a



Step response (unite change of velocity measurement; check the output of aero-drag)



(a) estimate of velocity

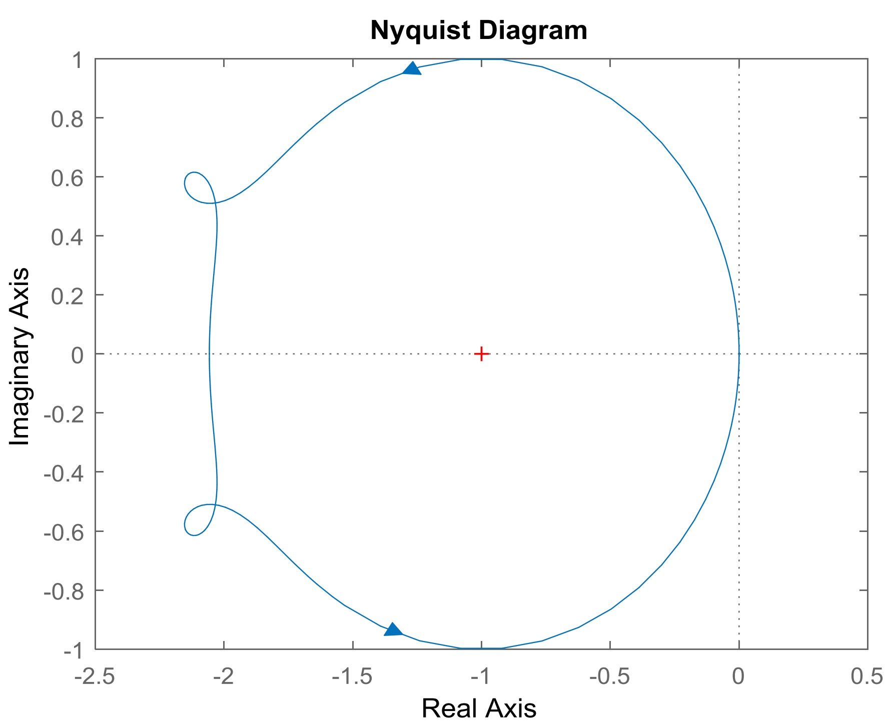


(b) estimate of aero-drag

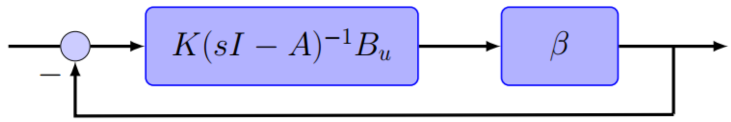
Q5.

Nyquist plot of the open-loop transfer function $L(s) = K(sI - A)^{-1}B$

```
looptf =

     1.486 s^3 + 2.68 s^2 + 0.3195 s + 0.1141
  -----------------------------------------------
  s^4 + 1.06 s^3 - 1.115 s^2 - 0.0658 s - 0.05547
```

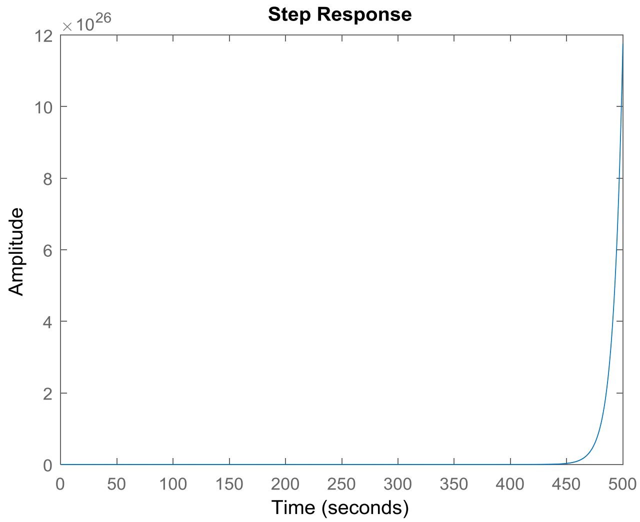


- Can study this by introducing a modification to the system, where nominally $\beta = 1$, but we would like to consider:
  - The gain $\beta \in \mathbb{R}$
  - The phase $\beta \in e^{j\phi}$

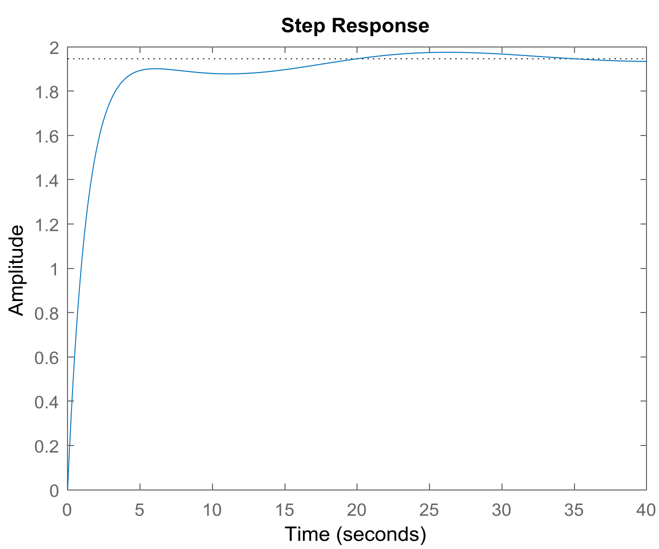


If $\beta \in (g_m, +\infty) = (0.4862, +\infty)$, the closed-loop system would be stable

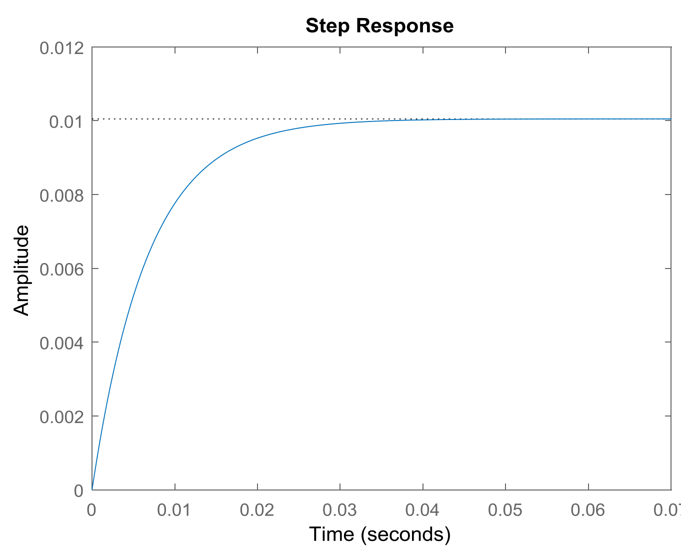
$\beta = 0.4$, unstable



$\beta = 1$, stable



$\beta = 100$, stable

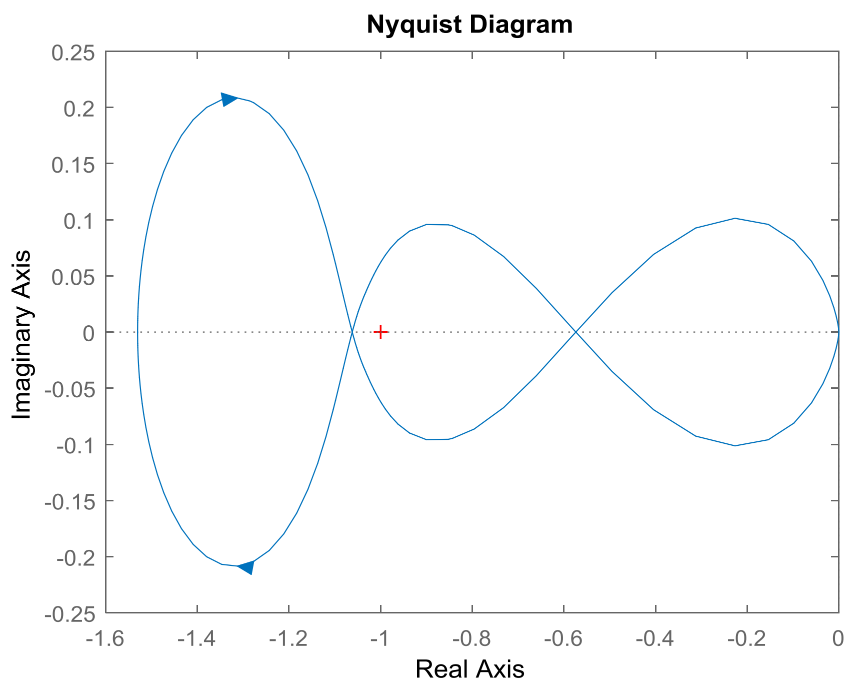


LQG transfer function from the measurement to the input

```
Glqg =

    -14.8 s^3 - 26.51 s^2 - 2.873 s - 1.123
  ---------------------------------------------
  s^4 + 4.028 s^3 + 6.435 s^2 + 3.735 s + 0.1158
```

Loop transfer function of LQG design

```
LoopLQG =

                 2.434 s^5 + 7.244 s^4 + 5.768 s^3 + 0.9765 s^2 + 0.244 s + 0.009828
  -------------------------------------------------------------------------------------------------
  s^8 + 5.088 s^7 + 9.591 s^6 + 5.999 s^5 - 3.423 s^4 - 4.69 s^3 - 0.7318 s^2 - 0.2148 s - 0.006422

Continuous-time transfer function.
```

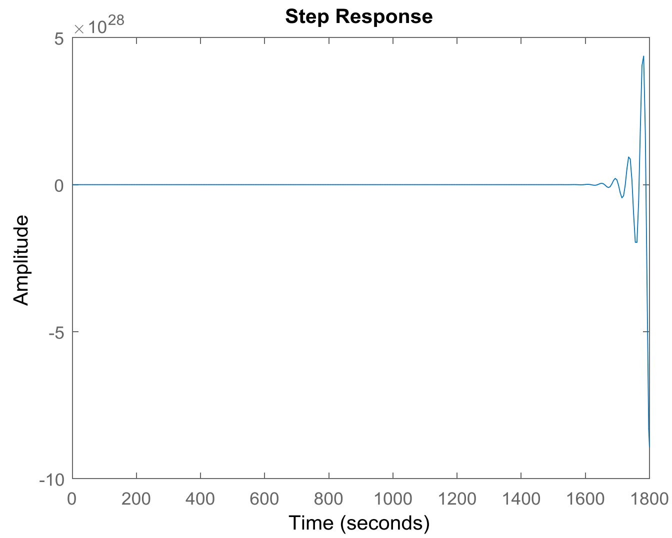


Open-loop poles (one unstable pole)

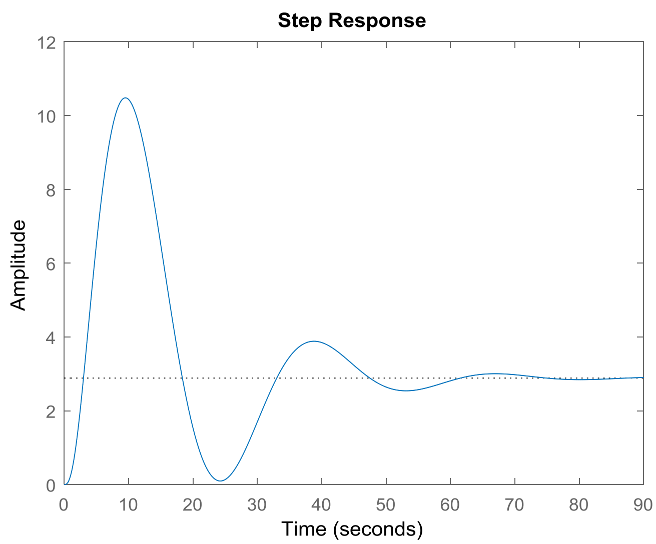
```
  -1.4034 + 0.9994i
  -1.4034 - 0.9994i
  -1.7036 + 0.0000i
  -1.1885 + 0.0000i
   0.7309 + 0.0000i
  -0.0438 + 0.2065i
  -0.0438 - 0.2065i
  -0.0328 + 0.0000i
```

If β ∈ $(g_{m1}, g_{m2}) = (0.9417, g_{m2})$, the closed-loop system would be stable

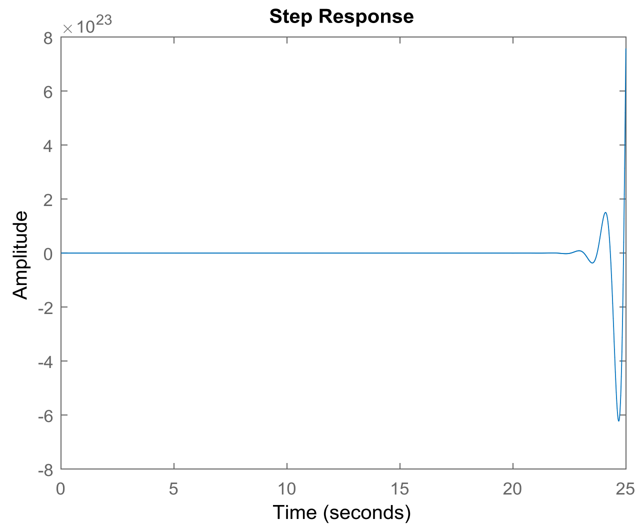
$\beta = 0.9$, unstable



$\beta = 1$, stable



$\beta = 100$, unstable



Q6

```
>> Kf{2}

ans =

  -44.0222
    4.9036
    6.8722
    3.7073
```

```
>> Kf{2}

ans =

  -44.0222
    4.9036
    6.8722
    3.7073
```

```
>> Kf{3}

ans =

  -46.0067
   11.5304
   18.2762
    6.0459
```

```
>> Kf{4}

ans =

  -52.7181
   30.3977
   54.3523
   10.4262
```

```
>> Kf{5}

ans =

  -71.5250
  117.8809
  236.3659
   21.7424
```

```matlab
%% Kalman filter
F = B;
q2 = [0,10,100,1000,2e4];
Kf = cell(length(q2),1);          % Kalman filter gain
LoopLQG = cell(length(q2),1);    % Loop transfer function of LQG controller
Gm = zeros(length(q2),1);         % Gain margin
Pm = zeros(length(q2),1);
for i = 1:length(q2)

    W = 0.01 ; V = 1+ q2(i);

    P = are(A',C'*W^(-1)*C,F*V*F');
    Kf{i} = P*C'*W^(-1);

    %% Open-loop transfer function of LQG design
    % L(s)=K*(sI - A + Kf*C + B*K)^(-1)*Kf*C(sI-A)^(-1)*B = sys2 * sys1 (co
    % sys2 = K*(sI - A + Kf*C + B*K)^(-1)*Kf;
    % sys1 = C(sI-A)^(-1)*B

    [Num1,Den1] = ss2tf(A-Kf{i}*C-B*G,Kf{i},G,0); % sys2
    [Num2,Den2] = ss2tf(A,B,C,0);             % sys1

    LoopLQG{i} = series(tf(Num1,Den1),tf(Num2,Den2));   %% loop transfer fu
    [Gm(i),Pm(i),~,~] = margin(LoopLQG{i});           %% Margins

end
```