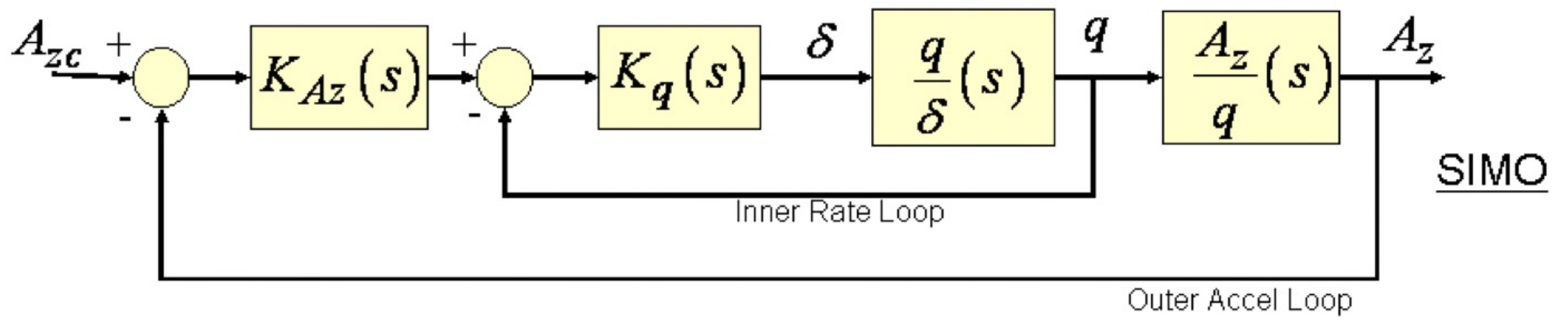




The state space model the for the aircraft is given as

$$\begin{bmatrix} \dot{\alpha} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} Z_\alpha/V & 1 \\ M_\alpha & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ q \end{bmatrix} + \begin{bmatrix} Z_\delta/V \\ M_\delta \end{bmatrix} \delta_e \quad \begin{bmatrix} A_z \\ q \end{bmatrix} = \begin{bmatrix} Z_\alpha & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha \\ q \end{bmatrix} + \begin{bmatrix} Z_\delta \\ 0 \end{bmatrix} \delta_e$$

where:  $Z_\alpha/V = -1.3046 \text{ (1/s)}$ ;  $Z_\delta/V = -0.2142 \text{ (1/s)}$ ;  $M_\alpha = 47.7109 \text{ (1/s}^2\text{)}$ ;  $M_\delta = -104.8346 \text{ (1/s}^2\text{)}$ ;  $V = 886.78 \text{ (fps)}$ ;

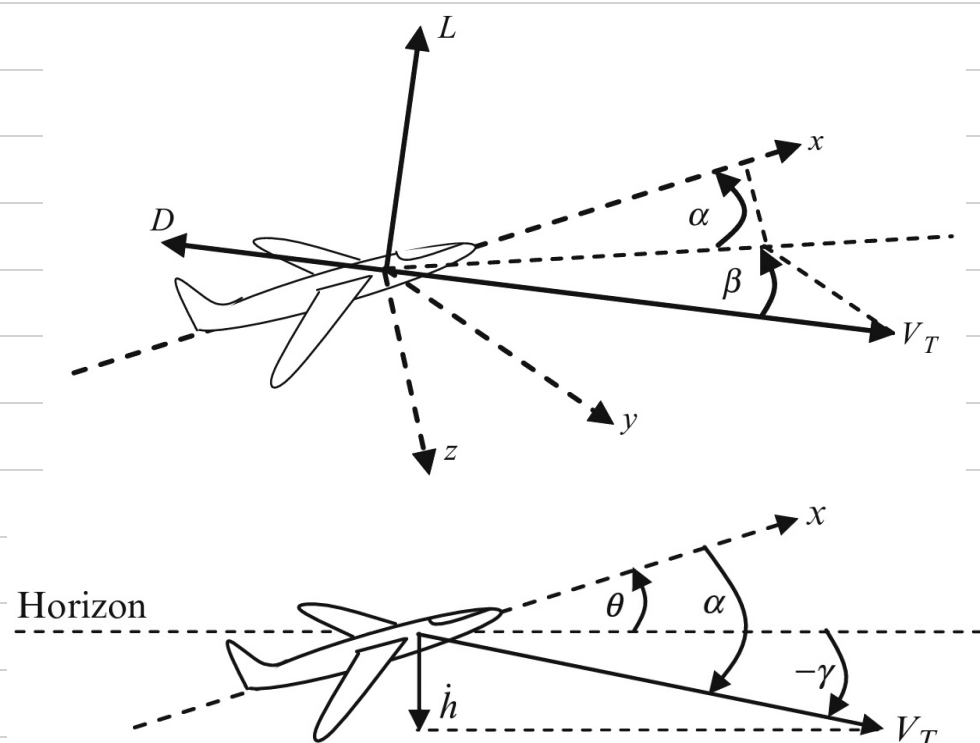
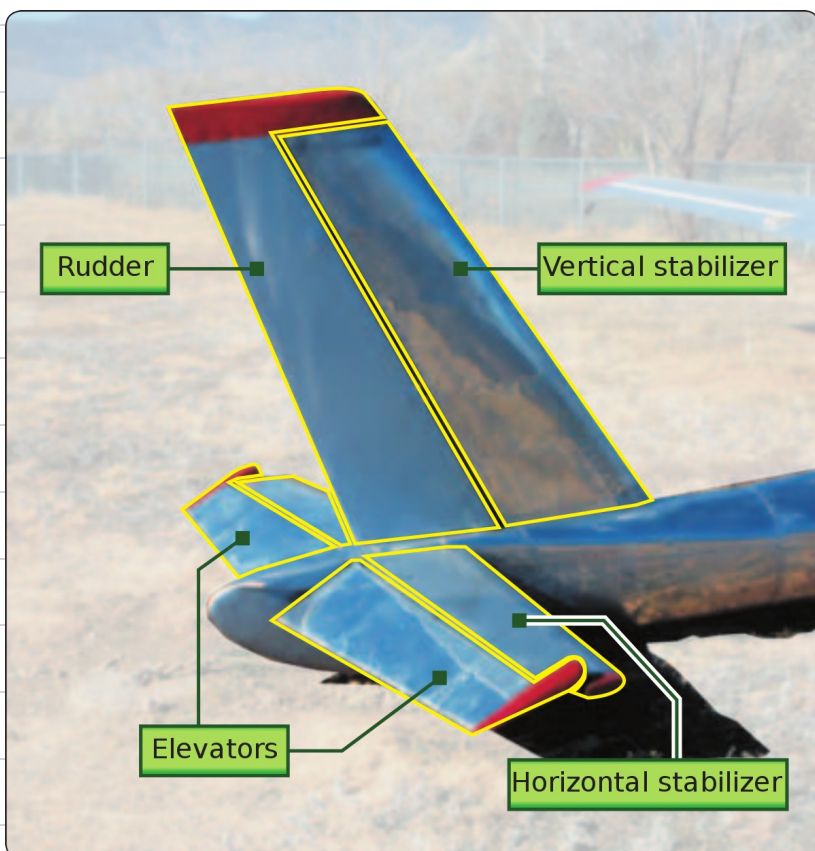
The autopilot (the controller) is a classical PI controller where

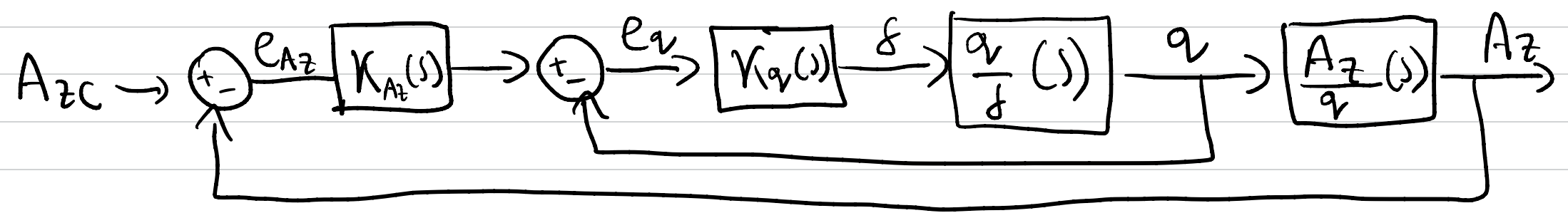
$$K_{Az}(s) = \frac{K_a(s+a_z)}{s} \quad K_q(s) = \frac{K_q(s+a_q)}{s}$$

where  $K_a = -0.0015$ ;  $K_q = -0.32$ ;  $a_z = 2.0$ ;  $a_q = 6.0$ ;

Background: This is a control system for the pitch of a plane

$\alpha$  = angle-of-attack,  $\dot{\alpha}$  = angle-of-attack rate,  $M_\alpha$  = Pitching moment coefficient  
 $q$  = Pitch rate,  $\dot{q}$  = Pitch acceleration,  $V$  = air speed  
 $A_z$  = Acceleration along z-axis,  $M_\delta$  = Pitch control derivative, sensitivity of aircraft to elevator input  
 $Z_\alpha$  = aerodynamic coefficient, change of lift/change in  $\alpha$ ,  $dL/d\alpha$ ,  $Z_\delta = dL/d\delta$   
 $\delta_e$  = deflection of the elevator control surface, at back of aircraft, controls  $\alpha, q$





i) Build state space Model for the Controller:

$$K_{Az}(s) = \frac{K_a(s+a_z)}{s}, \quad K_q(s) = \frac{K_q(s+a_q)}{s}$$

Controller State-Space Form:

two controllers  $\Rightarrow$  two states

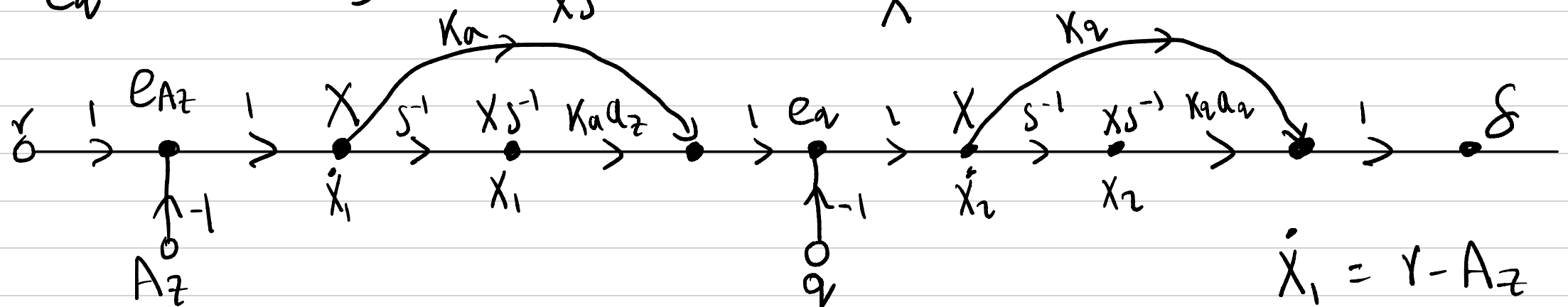
$$\begin{aligned} \dot{X}_c &= A_c X_c + B_{c1} Y + B_{c2} Y \\ U &= C_c X_c + D_{c1} Y + D_{c2} Y \end{aligned}$$

$$\left( (A_{zc} - A_z) K_{Az}(s) - q \right) K_q(s) = (A_{zc} - A_z) \cdot K_{Az}(s) \cdot K_q(s) - q K_q(s)$$

$$\delta_c = \frac{K_a(s+a_z)}{s} \cdot \frac{K_q(s+a_q)}{s} (A_{zc} - A_z) - \frac{K_q(s+a_q)}{s} q, \quad Y = A_{zc}, \quad Y = \begin{bmatrix} A_z \\ q \end{bmatrix}$$

$$\frac{K_{Az}(s)}{e_{Az}} = \frac{K_a(s+a_z)}{s} \cdot \frac{Xs^{-1}}{Xs^{-1}} = \frac{XK_a + XK_a a_z s^{-1}}{X}$$

$$\frac{K_q(s)}{e_q} = \frac{K_q(s+a_q)}{s} \cdot \frac{Xs^{-1}}{Xs^{-1}} = \frac{XK_q + XK_q a_q s^{-1}}{X}$$



$$\dot{x}_1 = r - A_z$$

$$\dot{x}_2 = K_a a_z x_1 + K_a \dot{x}_1 - q = K_a a_z x_1 + K_a (r - A_z) - q = K_a a_z x_1 + K_a r - K_a A_z - q$$

$$\delta = K_q \dot{x}_2 + K_q a_q x_2 = K_q (K_a a_z x_1 + K_a r - K_a A_z - q) + K_q a_q x_2$$

$$U = K_q K_a a_z x_1 + K_q K_a r - K_q K_a A_z - K_q q + K_q a_q x_2$$

$$\begin{aligned} \dot{X}_c &= \begin{bmatrix} 0 & 0 \\ K_a a_z & 0 \end{bmatrix} X_c + \begin{bmatrix} -1 & 0 \\ -K_a & -1 \end{bmatrix} \begin{bmatrix} A_z \\ q \end{bmatrix} + \begin{bmatrix} 1 \\ K_a \end{bmatrix} r \\ \delta_e &= \begin{bmatrix} K_q K_a a_z & K_q a_q \end{bmatrix} X_c + \begin{bmatrix} -K_q K_a & -K_q \end{bmatrix} \begin{bmatrix} A_z \\ q \end{bmatrix} + \begin{bmatrix} K_q K_a \end{bmatrix} r \end{aligned}$$

## ii) Build the closed Loop System:

### Plant State-Space Model:

$$\begin{bmatrix} \dot{\alpha} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} Z_\alpha/V & 1 \\ M_\alpha & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ q \end{bmatrix} + \begin{bmatrix} Z_\delta/V \\ M_\delta \end{bmatrix} \delta_e + w$$

$$\begin{bmatrix} \dot{A_z} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} Z_\alpha & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha \\ q \end{bmatrix} + \begin{bmatrix} Z_\delta \\ 0 \end{bmatrix} \delta_e + v$$

### Controller State-Space Model:

$$\dot{X}_c = \begin{bmatrix} 0 & 0 \\ K_a A_z & 0 \end{bmatrix} X_c + \begin{bmatrix} -1 & 0 \\ -K_a & -1 \end{bmatrix} \begin{bmatrix} A_z \\ q \end{bmatrix} + \begin{bmatrix} 1 \\ K_a \end{bmatrix} r$$

$$\delta_e = \begin{bmatrix} K_q K_a A_z & K_q A_q \end{bmatrix} X_c + \begin{bmatrix} -K_q K_a & -K_q \end{bmatrix} \begin{bmatrix} A_z \\ q \end{bmatrix} + \begin{bmatrix} K_q K_a \end{bmatrix} r$$

$$y = C_p X + D_p u \rightarrow u = C_c X_c + D_{c1} y + D_{c2} r \rightarrow u = C_c X_c + D_{c1} (C_p X + D_p u) + D_{c2} r$$

$$I u = C_c X_c + D_{c1} C_p X + D_{c1} D_p u + D_{c2} r \rightarrow \underbrace{(I - D_{c1} D_p)}_z u = C_c X_c + D_{c1} C_p X + D_{c2} r$$

$$u = z^{-1} C_c X_c + z^{-1} D_{c1} C_p X + z^{-1} D_{c2} r$$

$$\dot{X} = A_p X + B_p (z^{-1} C_c X_c + z^{-1} D_{c1} C_p X + z^{-1} D_{c2} r)$$

$$\text{Closed loop State Dynamics: } \dot{X} = (A_p + z^{-1} D_{c1} C_p) X + B_p z^{-1} C_c X_c + B_p z^{-1} D_{c2} r$$

$$\dot{X}_c = A_c X_c + B_{c1} y + B_{c2} r =$$

$$\dot{X}_c = A_c X_c + B_{c1} (C_p X + D_p (z^{-1} C_c X_c + z^{-1} D_{c1} C_p X + z^{-1} D_{c2} r)) + B_{c2} r$$

### Closed loop Controller model:

$$\dot{X}_c = (B_{c1} (I + D_p z^{-1} D_{c1}) C_p) X + (A_c + B_{c1} D_p z^{-1} C_c) X_c + (B_{c2} + B_{c1} D_p z^{-1} D_{c2}) r$$

$$\text{Output Equation: } y = C_p X + D_p u$$

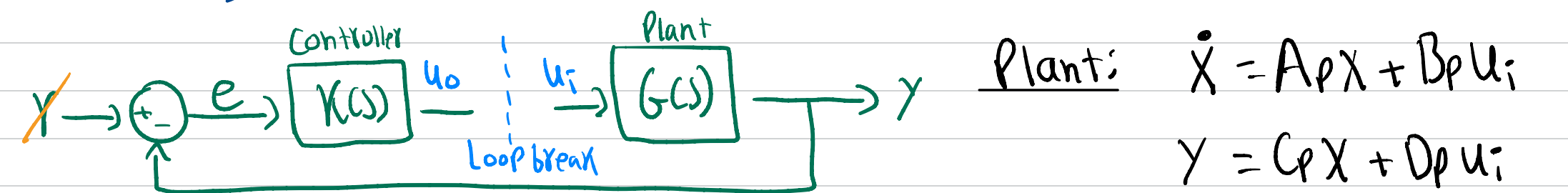
$$y = C_p X + D_p (z^{-1} C_c X_c + z^{-1} D_{c1} C_p X + z^{-1} D_{c2} r)$$

## Complete Closed Loop Model:

$$\begin{bmatrix} \dot{X} \\ \dot{X}_c \end{bmatrix} = \begin{bmatrix} A_p + B_p Z^{-1} D_{c1} C_p & B_p Z^{-1} C_c \\ B_{c1} (I + D_p Z^{-1} D_{c1}) C_p & A_c + B_{c1} D_p Z^{-1} C_c \end{bmatrix} \begin{bmatrix} X \\ X_c \end{bmatrix} + \begin{bmatrix} B_p Z^{-1} D_{c2} \\ B_{c2} + B_{c1} D_p Z^{-1} D_{c2} \end{bmatrix} Y$$

$$Y = \begin{bmatrix} (I + D_p Z^{-1} D_{c1}) C_p & D_p Z^{-1} C_c \end{bmatrix} \begin{bmatrix} X \\ X_c \end{bmatrix} + \begin{bmatrix} D_p Z^{-1} D_{c2} \end{bmatrix} Y$$

### iii.) Finding the Loop Transfer Function $L_u$ at the Plant Input:



- For stability analysis command  $Y$  is ignored

Controller:  $\dot{X}_c = A_c X_c + B_{c1} Y + B_{c2} Y$   
 $u_0 = C_c X_c + D_{c1} Y + D_{c2} Y$

$$\dot{X}_c = A_c X_c + B_{c1} (C_p X + D_p u_i) = B_{c1} C_p X + A_c X_c + B_{c1} D_p u_i$$

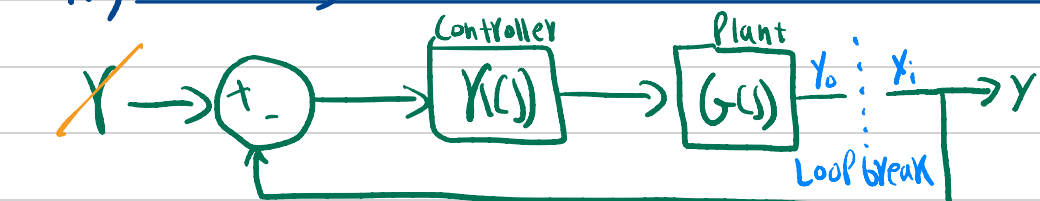
$$u_0 = C_c X_c + D_{c1} (C_p X + D_p u_i) = D_{c1} C_p X + C_c X_c + D_{c1} D_p u_i$$

$L_u(s)$ :

$$\begin{bmatrix} \dot{X} \\ \dot{X}_c \end{bmatrix} = \begin{bmatrix} A_p & 0 \\ B_{c1} C_p & A_c \end{bmatrix} \begin{bmatrix} X \\ X_c \end{bmatrix} + \begin{bmatrix} B_p \\ B_{c1} D_p \end{bmatrix} u_{in}$$

$$u_{out} = - \begin{bmatrix} D_{c1} C_p & C_c \end{bmatrix} \begin{bmatrix} X \\ X_c \end{bmatrix} - \begin{bmatrix} D_{c1} D_p \end{bmatrix} u_{in}$$

### iii.) Finding the Loop Transfer Function $L_y$ at the Plant Output:



$$\dot{X} = A_p X + B_p (C_c X_c + D_{c1} Y_i) = A_p X + B_p C_c X_c + B_p D_{c1} Y_i$$

$$Y_{out} = C_p X + D_p (C_c X_c + D_{c1} Y_i) = C_p X + D_p C_c X_c + D_p D_{c1} Y_i$$

$L_y(s)$ :

$$\begin{bmatrix} \dot{X} \\ \dot{X}_c \end{bmatrix} = \begin{bmatrix} A_p & B_p C_c \\ 0 & A_c \end{bmatrix} \begin{bmatrix} X \\ X_c \end{bmatrix} + \begin{bmatrix} B_p D_{c1} \\ B_{c1} \end{bmatrix} Y_{in}$$

$$Y_{out} = - \begin{bmatrix} C_p & D_p C_c \end{bmatrix} \begin{bmatrix} X \\ X_c \end{bmatrix} - \begin{bmatrix} D_p D_{c1} \end{bmatrix} Y_{in}$$

Controller:  $\dot{X}_c = A_c X_c + B_{c1} Y_i + B_{c2} Y$   
 $u = C_c X_c + D_{c1} Y_i + D_{c2} Y$



# SIMO Aircraft Problem, Pitch Control System

```
% Initializing symbolic Variables
syms alpha_dot q_dot Z_alpha M_alpha V alpha q Z_delta M_delta delta_e ...
a_z a_q K_q K_a alpha_dot_c q_dot_c alpha_c q_c
```

## Building the Plant State Space Model

```
A_p = [Z_alpha/V 1; M_alpha 0];
B_p = [Z_delta/V; M_delta];
C_p = [Z_alpha 0; 0 1];
D_p = [Z_delta; 0];
X_dot = [alpha_dot; q_dot];
X = [alpha; q];
y = [a_z; q];
u = delta_e;
% Displaying the Plant State Space Model
s = ["X_dot = A_p*X + B_p*u"; "y = C_p*X + D_p*u"];
displayFormula(["'Plant State Space Model:'"; s]);
```

Plant State Space Model:

$$\begin{pmatrix} \dot{\alpha} \\ \dot{q} \end{pmatrix} = \begin{pmatrix} \frac{Z_\alpha}{V} & 1 \\ M_\alpha & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ q \end{pmatrix} + \begin{pmatrix} \frac{Z_\delta}{V} \\ M_\delta \end{pmatrix} \delta_e$$

$$\begin{pmatrix} a_z \\ q \end{pmatrix} = \begin{pmatrix} Z_\alpha & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \alpha \\ q \end{pmatrix} + \begin{pmatrix} Z_\delta \\ 0 \end{pmatrix} \delta_e$$

## 1.1.i) Building the Controller State Space Model

```
A_c = [0 0; K_a*a_z 0];
B_c1 = [-1 0; -K_a -1];
B_c2 = [1; K_a];
C_c = [K_q*K_a*a_z K_q*a_q];
D_c1 = [-K_q*K_a -K_q];
D_c2 = [K_q*K_a];
X_dot_c = [alpha_dot_c; q_dot_c];
X_c = [alpha_c; q_c];
s = ["X_dot_c = A_c*X_c + B_c1*y + B_c2*r"; "delta_e = C_c*X_c + D_c1*y + D_c2*r"];
displayFormula(["'Controller State Space Model:'"; s]);
```

Controller State Space Model:

$$\begin{pmatrix} \dot{\alpha}_c \\ \dot{q}_c \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ K_a a_z & 0 \end{pmatrix} \begin{pmatrix} \alpha_c \\ q_c \end{pmatrix} + \begin{pmatrix} -1 & 0 \\ -K_a & -1 \end{pmatrix} \begin{pmatrix} a_z \\ q \end{pmatrix} + \begin{pmatrix} 1 \\ K_a \end{pmatrix} r$$

$$\delta_e = \begin{pmatrix} K_a K_q a_z & K_q a_q \end{pmatrix} \begin{pmatrix} \alpha_c \\ q_c \end{pmatrix} + \begin{pmatrix} -K_a K_q & -K_q \end{pmatrix} \begin{pmatrix} a_z \\ q \end{pmatrix} + K_a K_q r$$

## 1.1.ii) Building the Closed Loop State Space Model

```

I = eye(size(D_c1*D_p));
z = (I-D_c1*D_p);
A_cl = [A_p+(B_p*(inv(z))*D_c1*C_p) B_p*inv(z)*C_c;...
B_c1*(eye(size(D_p*inv(z)*D_c1))+D_p*inv(z)*D_c1)*C_p A_c+B_c1*D_p*inv(z)*C_c];
B_cl = [B_p*inv(z)*D_c2;B_c2+B_c1*D_p*inv(z)*D_c2];
C_cl = [(eye(size(D_p*inv(z)*D_c1))+D_p*inv(z)*D_c1)*C_p D_p*inv(z)*C_c];
D_cl = [D_p*inv(z)*D_c2];
X_dot_cl = [X_dot;X_dot_c];
X_cl = [X;X_c];
s = ["X_dot_cl = A_cl*X_cl + B_cl*r";"y= C_cl*X_cl + D_cl*r"];
displayFormula(["'Complete Closed Loop State Space Model:'";s]);

```

Complete Closed Loop State Space Model:

$$\begin{pmatrix} \dot{\alpha} \\ \dot{q} \\ \dot{\alpha}_c \\ \dot{q}_c \end{pmatrix} = \begin{pmatrix} \frac{Z_\alpha}{V} - \frac{K_a K_q Z_\alpha Z_\delta}{V \sigma_2} & 1 - \frac{K_q Z_\delta}{V \sigma_2} & \frac{K_a K_q Z_\delta a_z}{V \sigma_2} & \frac{K_q Z_\delta a_q}{V \sigma_2} \\ M_\alpha - \frac{K_a K_q M_\delta Z_\alpha}{\sigma_2} & -\frac{K_q M_\delta}{\sigma_2} & \frac{K_a K_q M_\delta a_z}{\sigma_2} & \frac{K_q M_\delta a_q}{\sigma_2} \\ Z_\alpha (\sigma_1 - 1) & \frac{K_q Z_\delta}{\sigma_2} & -\frac{K_a K_q Z_\delta a_z}{\sigma_2} & -\frac{K_q Z_\delta a_q}{\sigma_2} \\ K_a Z_\alpha (\sigma_1 - 1) & \sigma_1 - 1 & K_a a_z - \frac{K_a^2 K_q Z_\delta a_z}{\sigma_2} & -\frac{K_a K_q Z_\delta a_q}{\sigma_2} \end{pmatrix} \begin{pmatrix} \alpha \\ q \\ \alpha_c \\ q_c \end{pmatrix} + \begin{pmatrix} \frac{K_a K_q Z_\delta}{V \sigma_2} \\ \frac{K_a K_q M_\delta}{\sigma_2} \\ 1 - \sigma_1 \\ K_a - \frac{K_a^2 K_q Z_\delta}{\sigma_2} \end{pmatrix} r$$

where

$$\sigma_1 = \frac{K_a K_q Z_\delta}{\sigma_2}$$

$$\sigma_2 = K_a K_q Z_\delta + 1$$

$$\begin{pmatrix} a_z \\ q \end{pmatrix} = \begin{pmatrix} -Z_\alpha \left( \frac{K_a K_q Z_\delta}{\sigma_1} - 1 \right) & -\frac{K_q Z_\delta}{\sigma_1} & \frac{K_a K_q Z_\delta a_z}{\sigma_1} & \frac{K_q Z_\delta a_q}{\sigma_1} \\ 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ q \\ \alpha_c \\ q_c \end{pmatrix} + \begin{pmatrix} \frac{K_a K_q Z_\delta}{\sigma_1} \\ 0 \end{pmatrix} r$$

where

$$\sigma_1 = K_a K_q Z_\delta + 1$$

## 1.1.ii) Subbing in the Numerical Values to the Closed Loop Model

```

% V = 886.78; % Feet per second
% Z_alpha = -1.3046*V;
% Z_delta = -0.2142*V;
% M_alpha = 47.7109;
% M_delta = -104.8346;
% K_a = -0.0015;
% K_q = -0.32;
% a_z = 2.0;
% a_q = 6.0;
variables = [V;Z_alpha;Z_delta;M_alpha;M_delta;K_a;K_q;a_z;a_q];
variable_values = [886.78;-1.3046*886.78;-0.2142*886.78;47.7109;-104.8346;-0.0015;-0.32;2.0;6.0];
A_cl = double(subs(A_cl,variables,variable_values));
B_cl = double(subs(B_cl,variables,variable_values));
C_cl = double(subs(C_cl,variables,variable_values));
D_cl = double(subs(D_cl,variables,variable_values));
A_cl0 = round(vpa(A_cl),2);
B_cl0 = round(vpa(B_cl),2);
C_cl0 = round(vpa(C_cl),2);
D_cl0 = round(vpa(D_cl),2);

s = ["X_dot_cl = A_cl0*X_cl + B_cl0*r";"y= C_cl0*X_cl + D_cl0*r"];
displayFormula(["'Complete Closed Loop State Space Model:'";s]);

```

Complete Closed Loop State Space Model:

$$\begin{pmatrix} \dot{\alpha} \\ \dot{q} \\ \dot{\alpha}_c \\ \dot{q}_c \end{pmatrix} = \begin{pmatrix} -1.44 & 0.92 & 0 & 0.45 \\ -16.34 & -36.91 & -0.11 & 221.48 \\ 1272.96 & 66.88 & 0.2 & -401.29 \\ -1.91 & -1.1 & 0 & 0.6 \end{pmatrix} \begin{pmatrix} \alpha \\ q \\ \alpha_c \\ q_c \end{pmatrix} + \begin{pmatrix} 0 \\ -0.06 \\ 1.1 \\ 0 \end{pmatrix} r$$

$$\begin{pmatrix} a_z \\ q \end{pmatrix} = \begin{pmatrix} -1272.96 & -66.88 & -0.2 & 401.29 \\ 0 & 1.0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ q \\ \alpha_c \\ q_c \end{pmatrix} + \begin{pmatrix} -0.1 \\ 0 \end{pmatrix} r$$

```

A_cl_eig = double(eig(A_cl));
disp(["Eigenvalues of Closed Loop System: "; num2str(A_cl_eig)]);

```

```

"Eigenvalues of Closed Loop System: "
"-1.81029+0i      "
"-3.65126+1.64868i"
"-3.65126-1.64868i"
"-28.4327+0i      "

```

### 1.1.iii) Loop Transfer Function at the Plant Input L\_u

```

A_Lu = [A_p 0*A_p;B_c1*C_p A_c];
B_Lu = [B_p;B_c1*D_p];
C_Lu = -[D_c1*C_p C_c];
D_Lu = -[D_c1*D_p];
s = ["X_dot_cl = A_Lu*X_cl + B_Lu*u_in";"u_out= C_Lu*X_cl + D_Lu*u_in"];

```

```
displayFormula(['Loop Opened at Plant Input:'];s]);
```

Loop Opened at Plant Input:

$$\begin{pmatrix} \dot{\alpha} \\ \dot{q} \\ \dot{\alpha}_c \\ \dot{q}_c \end{pmatrix} = \begin{pmatrix} \frac{Z_\alpha}{V} & 1 & 0 & 0 \\ M_\alpha & 0 & 0 & 0 \\ -Z_\alpha & 0 & 0 & 0 \\ -K_a Z_\alpha & -1 & K_a a_z & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ q \\ \alpha_c \\ q_c \end{pmatrix} + \begin{pmatrix} \frac{Z_\delta}{V} \\ M_\delta \\ -Z_\delta \\ -K_a Z_\delta \end{pmatrix} u_{in}$$

$$u_{out} = \begin{pmatrix} K_a K_q Z_\alpha & K_q & -K_a K_q a_z & -K_q a_q \end{pmatrix} \begin{pmatrix} \alpha \\ q \\ \alpha_c \\ q_c \end{pmatrix} + K_a K_q Z_\delta u_{in}$$

```
A_Lu = double(subs(A_Lu,variables,variable_values));
B_Lu = double(subs(B_Lu,variables,variable_values));
C_Lu = double(subs(C_Lu,variables,variable_values));
D_Lu = double(subs(D_Lu,variables,variable_values));
A_Lu0 = round(vpa(A_Lu),2);
B_Lu0 = round(vpa(B_Lu),2);
C_Lu0 = round(vpa(C_Lu),2);
D_Lu0 = round(vpa(D_Lu),2);
s = ["X_dot_cl = A_Lu0*X_cl + B_Lu0*u_in";"u_out= C_Lu0*X_cl + D_Lu0*u_in"];
displayFormula(['Loop Opened at Plant Input:'];s]);
```

Loop Opened at Plant Input:

$$\begin{pmatrix} \dot{\alpha} \\ \dot{q} \\ \dot{\alpha}_c \\ \dot{q}_c \end{pmatrix} = \begin{pmatrix} -1.3 & 1.0 & 0 & 0 \\ 47.71 & 0 & 0 & 0 \\ 1156.89 & 0 & 0 & 0 \\ -1.74 & -1.0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ q \\ \alpha_c \\ q_c \end{pmatrix} + \begin{pmatrix} -0.21 \\ -104.83 \\ 189.95 \\ -0.28 \end{pmatrix} u_{in}$$

$$u_{out} = \begin{pmatrix} -0.56 & -0.32 & 0 & 1.92 \end{pmatrix} \begin{pmatrix} \alpha \\ q \\ \alpha_c \\ q_c \end{pmatrix} - 0.09 u_{in}$$

```
sys_Lu = ss(A_Lu,B_Lu,C_Lu,D_Lu);
[b,a] = ss2tf(A_Lu,B_Lu,C_Lu,D_Lu);
sys_Lu_TF = tf(b,a)
```

sys\_Lu\_TF =

$$\frac{-0.09118 s^4 + 32.82 s^3 + 309.8 s^2 + 782.7 s + 750.8}{s^4 + 1.305 s^3 - 47.71 s^2}$$

Continuous-time transfer function.

## 1.1.iv) Loop Transfer Function at the Plant Output L\_y



```

A_Ly = [A_p B_p*C_c;0*A_p A_c];
B_Ly = [B_p*D_c1;B_c1];
C_Ly = -[C_p D_p*C_c];
D_Ly = -[D_p*D_c1];
s = ["X_dot_c1 = A_Ly*X_c1 + B_Ly*y_in";"y_out= C_Ly*X_c1 + D_Ly*y_in"];
displayFormula(['Loop Opened at Plant Output: ';s]);

```

Loop Opened at Plant Output:

$$\begin{pmatrix} \dot{\alpha} \\ \dot{q} \\ \dot{\alpha}_c \\ \dot{q}_c \end{pmatrix} = \begin{pmatrix} \frac{Z_\alpha}{V} & 1 & \frac{K_a K_q Z_\delta a_z}{V} & \frac{K_q Z_\delta a_q}{V} \\ M_\alpha & 0 & K_a K_q M_\delta a_z & K_q M_\delta a_q \\ 0 & 0 & 0 & 0 \\ 0 & 0 & K_a a_z & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ q \\ \alpha_c \\ q_c \end{pmatrix} + \begin{pmatrix} -\frac{K_a K_q Z_\delta}{V} & -\frac{K_q Z_\delta}{V} \\ -K_a K_q M_\delta & -K_q M_\delta \\ -1 & 0 \\ -K_a & -1 \end{pmatrix} y_{in}$$

$$y_{out} = \begin{pmatrix} -Z_\alpha & 0 & -K_a K_q Z_\delta a_z & -K_q Z_\delta a_q \\ 0 & -1 & 0 & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ q \\ \alpha_c \\ q_c \end{pmatrix} + \begin{pmatrix} K_a K_q Z_\delta & K_q Z_\delta \\ 0 & 0 \end{pmatrix} y_{in}$$

```

A_Ly = double(subs(A_Ly,variables,variable_values));
B_Ly = double(subs(B_Ly,variables,variable_values));
C_Ly = double(subs(C_Ly,variables,variable_values));
D_Ly = double(subs(D_Ly,variables,variable_values));
sys_Ly = ss(A_Ly,B_Ly,C_Ly,D_Ly);
A_Ly0 = round(vpa(A_Ly,2),2);
B_Ly0 = round(vpa(B_Ly,2),2);
C_Ly0 = round(vpa(C_Ly,2),2);
D_Ly0 = round(vpa(D_Ly,2),2);
s = ["X_dot_c1 = A_Ly0*X_c1 + B_Ly0*y_in";"y_out= C_Ly0*X_c1 + D_Ly0*y_in"];
displayFormula(['Loop Opened at Plant Output: ';s]);

```

Loop Opened at Plant Output:

$$\begin{pmatrix} \dot{\alpha} \\ \dot{q} \\ \dot{\alpha}_c \\ \dot{q}_c \end{pmatrix} = \begin{pmatrix} -1.3 & 1.0 & 0 & 0.41 \\ 47.71 & 0 & -0.1 & 201.28 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ q \\ \alpha_c \\ q_c \end{pmatrix} + \begin{pmatrix} 0 & -0.07 \\ 0.05 & -33.55 \\ -1.0 & 0 \\ 0 & -1.0 \end{pmatrix} y_{in}$$

$$y_{out} = \begin{pmatrix} 1156.89 & 0 & 0.18 & -364.7 \\ 0 & -1.0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ q \\ \alpha_c \\ q_c \end{pmatrix} + \begin{pmatrix} -0.09 & 60.78 \\ 0 & 0 \end{pmatrix} y_{in}$$

```

[b,a] = ss2tf(A_Lu,B_Lu,C_Lu,D_Lu);
sys_Ly_TF = tf(b,a)

```

sys\_Ly\_TF =

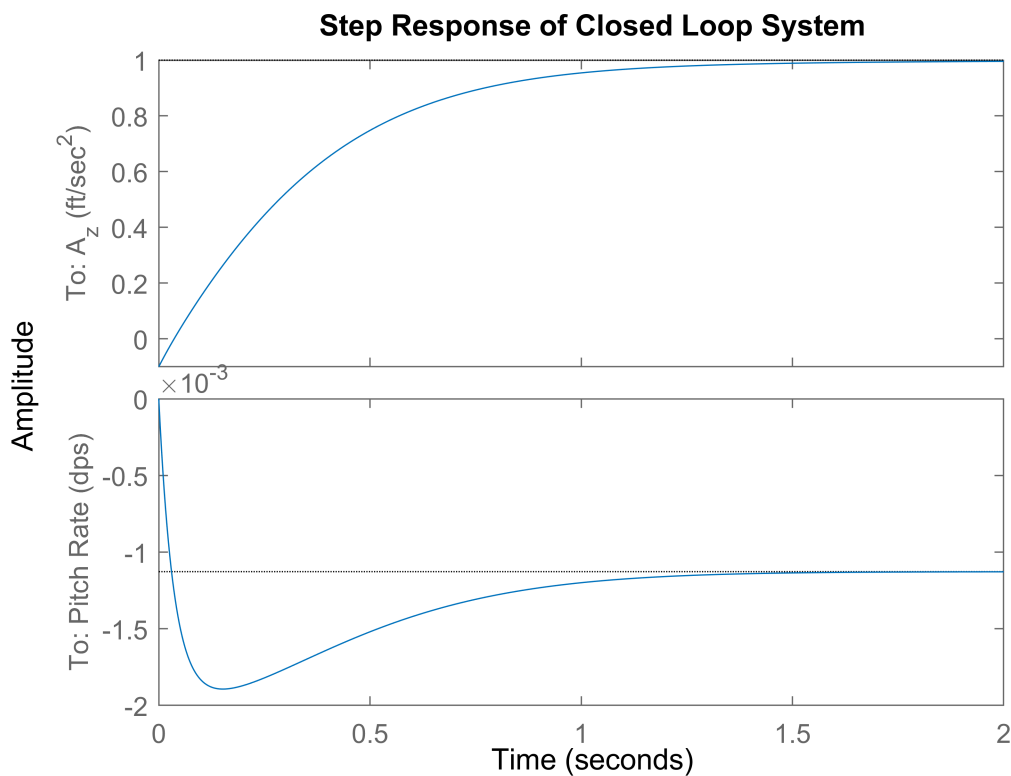
$$-0.09118 s^4 + 32.82 s^3 + 309.8 s^2 + 782.7 s + 750.8$$

$$s^4 + 1.305 s^3 - 47.71 s^2$$

Continuous-time transfer function.

## 1.2) Computing the Closed Loop System Step Response, Rise Time & Settling Time

```
sys_cl = ss(A_cl,B_cl,C_cl,D_cl);
sys_cl(1,1).OutputName = 'A_z (ft/sec^2)';
sys_cl(2,1).OutputName = 'Pitch Rate (dps)';
step(sys_cl)
xlim([0,2])
title("Step Response of Closed Loop System")
```



```
info = stepinfo(sys_cl, 'RiseTimeThreshold',[0 0.63], 'SettlingTimeThreshold',0.05)
```

```
info = 2x1 struct
```

...

| Fields | RiseTime | TransientTime | SettlingTime | SettlingMin | SettlingMax |
|--------|----------|---------------|--------------|-------------|-------------|
| 1      | 0.3457   | 0.9446        | 0.973        | 0.6301      | 0.9993      |
| 2      | 0.0159   | 1.061         | 1.061        | -0.0019     | -0.00072018 |

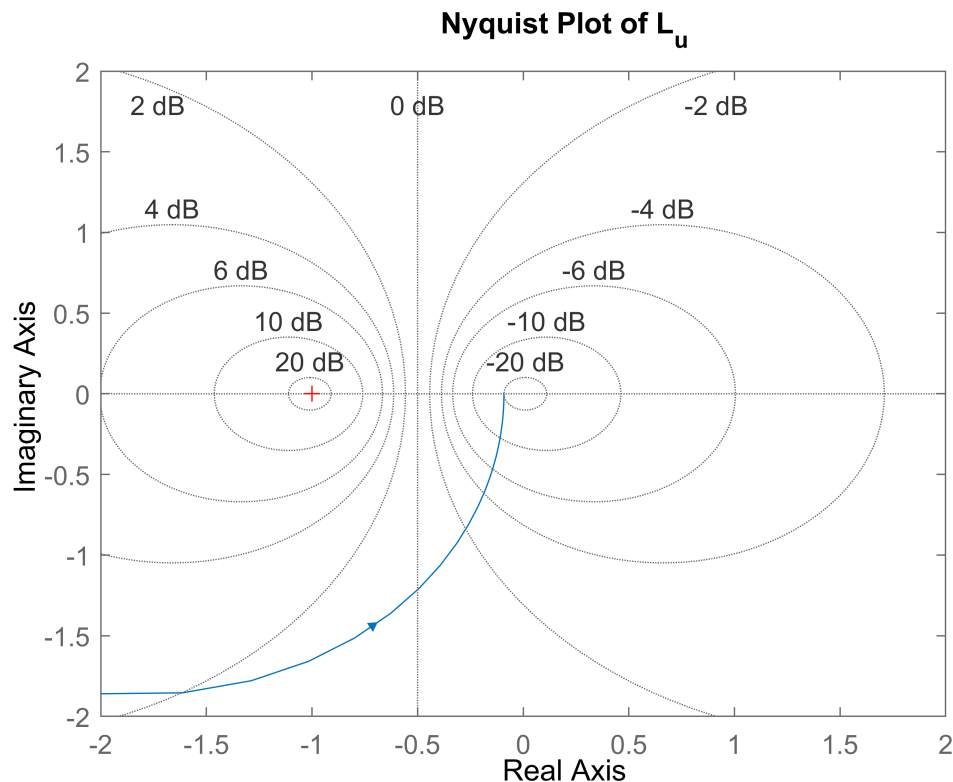
```
text = ['Rise Time of Az (63%): ', num2str(info(1).RiseTime), ' sec',...
newline 'Settling Time of Az (95%): ', num2str(info(1).SettlingTime), 'sec',...
newline 'Rise Time of q (63%): ', num2str(info(2).RiseTime), 'sec',...]
```

```
newline 'Settling Time of q (95%): ', num2str(info(2).SettlingTime), 'sec'];
disp(text)
```

```
Rise Time of Az (63%): 0.34573 sec,
Settling Time of Az (95%): 0.97305sec
Rise Time of q (63%): 0.015932sec,
Settling Time of q (95%): 1.061sec
```

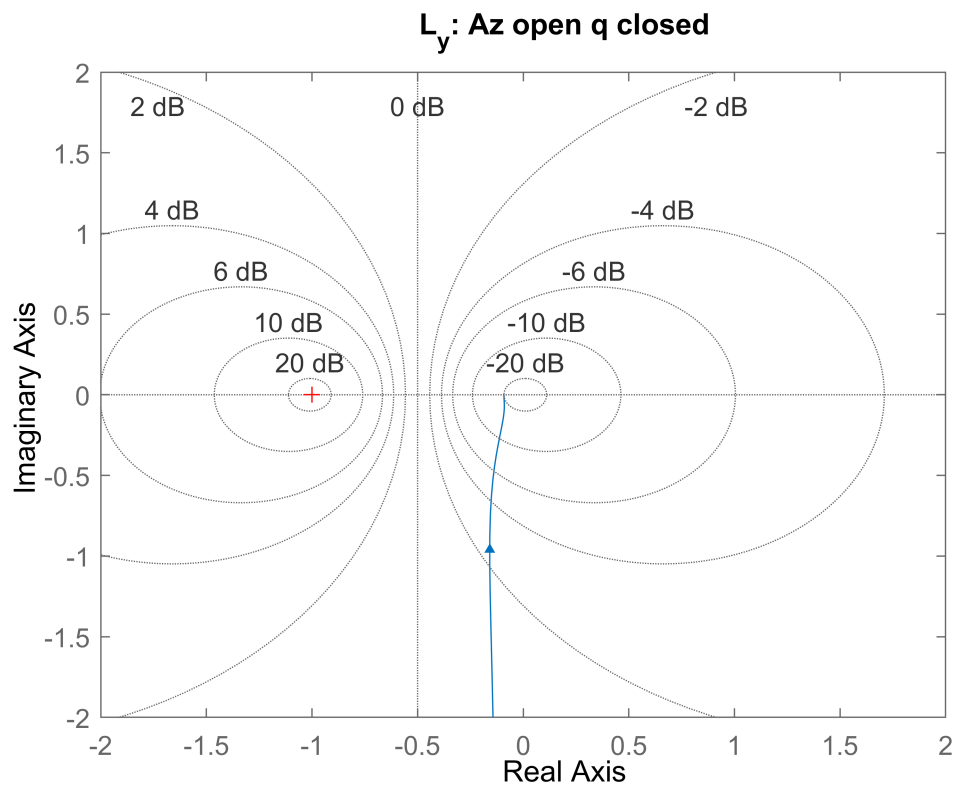
### 1.3.i) Nyquist Plot of $L_u$

```
h = nyquistplot(sys_Lu);
title("Nyquist Plot of  $L_u$ ")
setoptions(h, 'ShowFullContour', 'off');
grid on
xlim([-2,2])
ylim([-2,2])
```



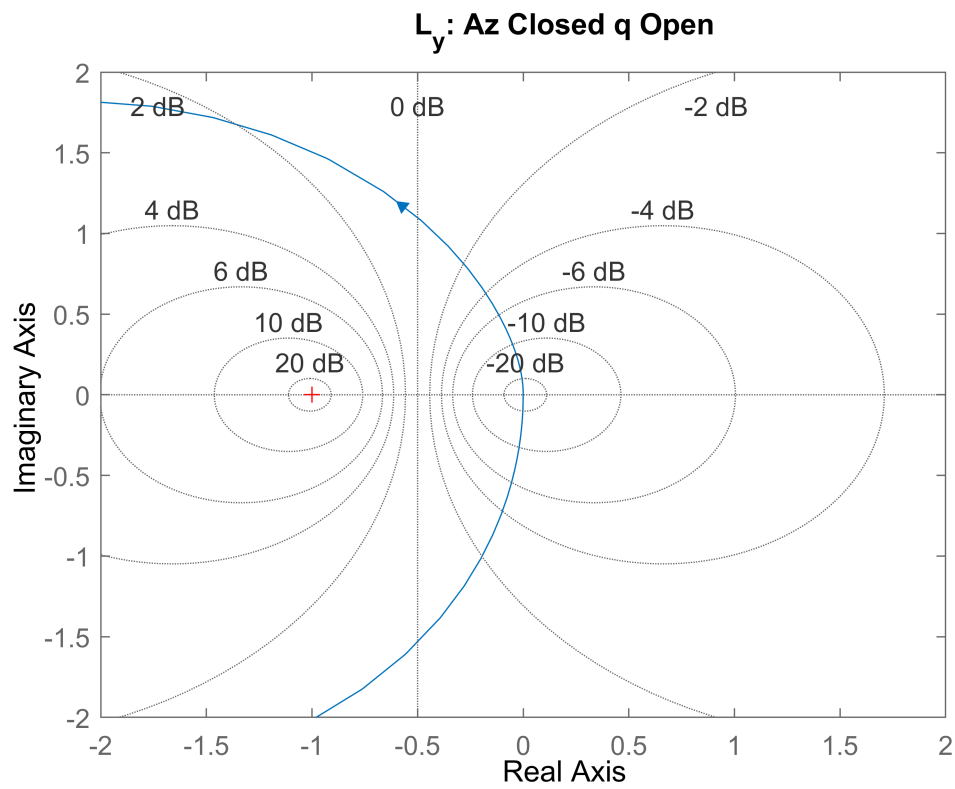
### 1.3.ii) Isolate Outputs: Az open q closed, Nyquist Plot

```
Ly_Az_Open = feedback(sys_Ly,eye(size(sys_Ly,1)-1),2,2);
h = nyquistplot(Ly_Az_Open(1,1));
setoptions(h, 'ShowFullContour', 'off');
grid on
xlim([-2,2])
ylim([-2,2])
title("L_y: Az open q closed")
```



### 1.3.iii) Isolate Outputs: Az Closed q Open, Nyquist Plot

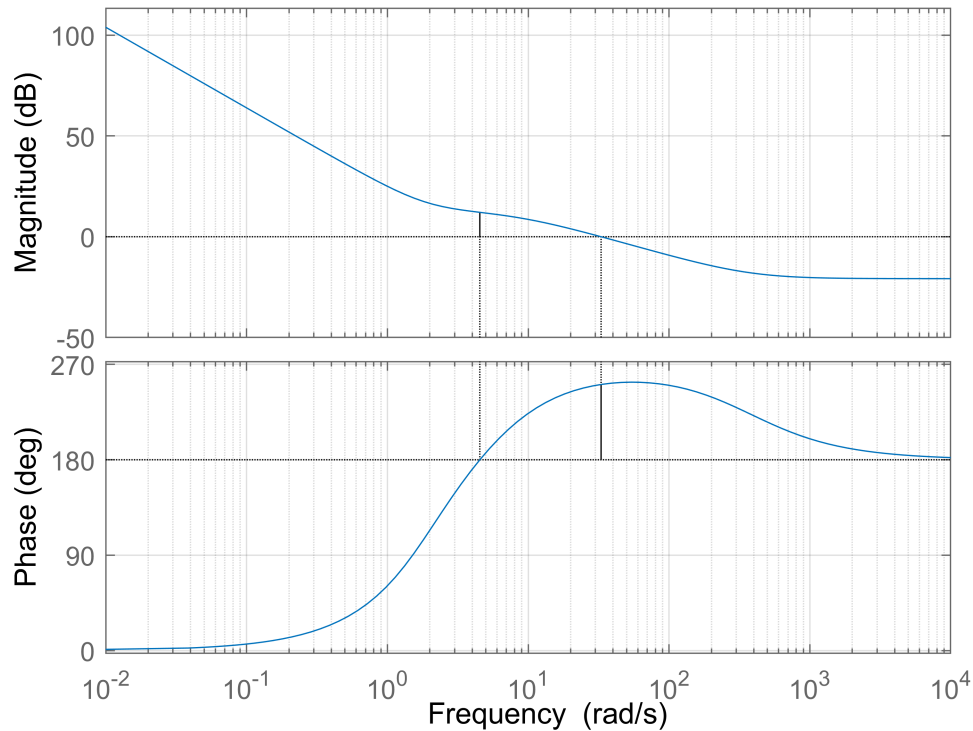
```
Ly_q_Open = feedback(sys_Ly,eye(size(sys_Ly,1)-1),1,1);
h = nyquistplot(Ly_q_Open(2,2));
setoptions(h,'ShowFullContour','off');
grid on
xlim([-2,2])
ylim([-2,2])
title(" $L_y$ : Az Closed q Open")
```



#### 1.4) Bode Plot for $L_u$

```
margin(sys_Lu)
grid on
title("Frequency Response at Plant Input ,  $L_u$ ")
```

### Frequency Response at Plant Input , $L_u$



```
allmargin(sys_Lu)
```

```
ans = struct with fields:
    GainMargin: [0.2481 10.9679]
    GMFrequency: [4.5341 Inf]
    PhaseMargin: 71.0668
    PMFrequency: 32.9302
    DelayMargin: 0.0377
    DMFrequency: 32.9302
    Stable: 1
```

```
lu = allmargin(sys_Lu);
text = ['Gain Margin: ', num2str(20*log10(lu.GainMargin(2))), ' dB',...
newline 'LGCF @: ', num2str(lu.PMFrequency), ' rad/s',...
newline 'Phase margin: ' num2str(lu.PhaseMargin), ' degrees',...
newline 'Phase Crossover @: ', num2str(lu.GMFrequency(1)), ' rad/s'];
disp(text)
```

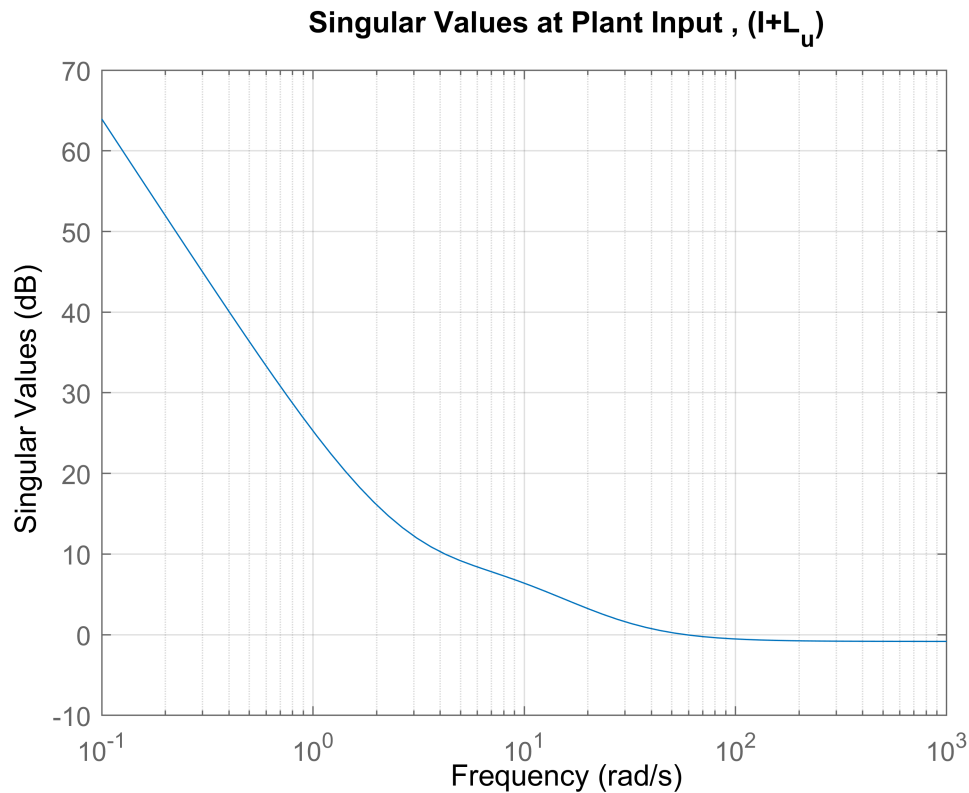
```
Gain Margin: 20.8025 dB,
LGCF @: 32.9302 rad/s
Phase margin: 71.0668 degrees,
Phase Crossover @: 4.5341 rad/s
```

## 1.5) Plotting the singular values for $(I+L_u)$ & Computing the Minimum Sigma

```
sigma(sys_Lu+1,{.1,1000})
sys_Lu_sigma_min = min(sigma(sys_Lu+1,{.1,1000}));
grid on
```



```
title("Singular Values at Plant Input , (I+L_u)")
```



```
fprintf("Minimum Sigma Value of (1+L_u): %d", sys_Lu_sigma_min);
```

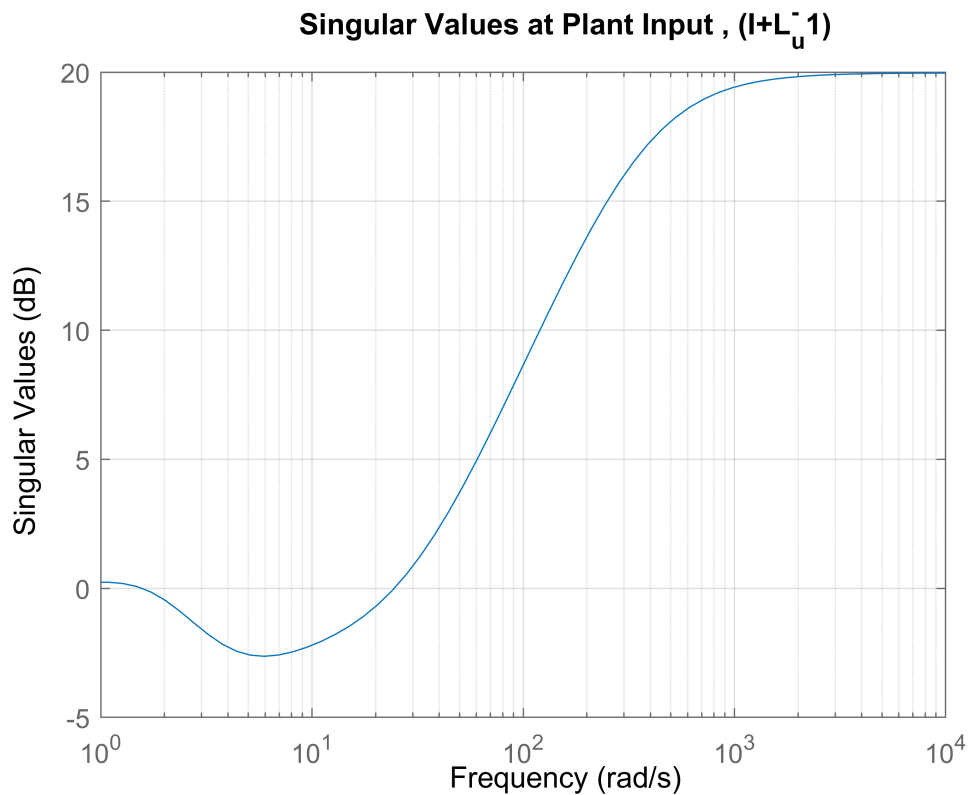
Minimum Sigma Value of (1+L\_u): 9.091591e-01

```
sys_Lu_sigma_min_db = 20*log10(sys_Lu_sigma_min);
fprintf("Minimum Sigma Value of (1+L_u) in db: %d", sys_Lu_sigma_min_db);
```

Minimum Sigma Value of (1+L\_u) in db: -8.272022e-01

## 1.6) Plotting the singular values for $(I+L_u^{-1})$ & Computing the Minimum Sigma

```
sigma(sys_Lu^-1+1)
sys_Lu_sigma1_min = min(sigma(sys_Lu^(-1)+1));
grid on
title("Singular Values at Plant Input , (I+L_u^-1)")
```



```
fprintf("Minimum Sigma Value of  $(1+L_u^{-1})$ : %d", sys_Lu_sigma1_min);
```

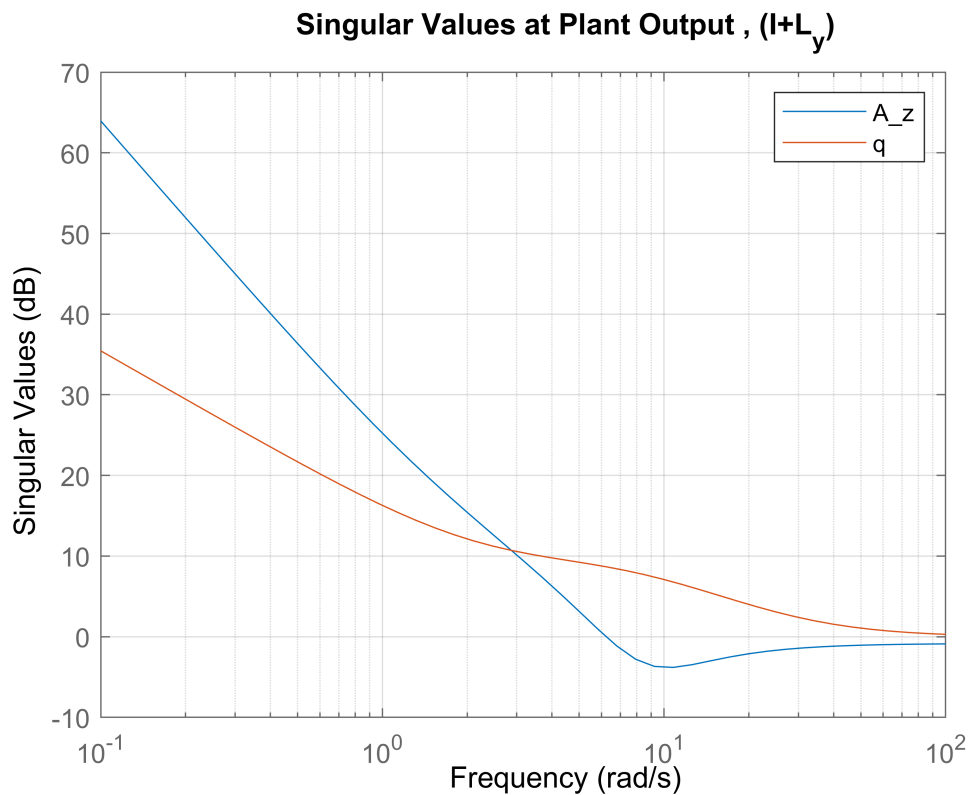
```
Minimum Sigma Value of  $(1+L_u^{-1})$ : 7.385623e-01
```

```
sys_Lu_sigma1_min_db = 20*log10(sys_Lu_sigma1_min);
fprintf("Minimum Sigma Value of  $(1+L_u)$  in db: %d", sys_Lu_sigma1_min_db);
```

```
Minimum Sigma Value of  $(1+L_u)$  in db: -2.632258e+00
```

## 1.7) Plotting the singular values for $(I+L_y)$

```
sigma(sys_Ly(1,1)+1,sys_Ly(2,2)+1)
sys_Ly_sigma_min_A_z = min(sigma(sys_Ly(1,1)+1));
sys_Ly_sigma_min_q = min(sigma(sys_Ly(2,2)+1));
grid on
title("Singular Values at Plant Output ,  $(I+L_y)$ ")
legend("A_z","q")
```



```
fprintf("Minimum Sigma Value of (1+L_y) for A_z: %d", sys_Ly_sigma_min_A_z);
```

Minimum Sigma Value of (1+L\_y) for A\_z: 6.441843e-01

```
sys_Ly_sigma_min_db = 20*log10(sys_Ly_sigma_min_A_z);
fprintf("Minimum Sigma Value of (1+L_y) for A_z in db: %d", sys_Ly_sigma_min_db);
```

Minimum Sigma Value of (1+L\_y) for A\_z in db: -3.819798e+00

```
fprintf("Minimum Sigma Value of (1+L_y) for q: %d", sys_Ly_sigma_min_q);
```

Minimum Sigma Value of (1+L\_y) for q: 1.034950e+00

```
sys_Ly_sigma_min_db = 20*log10(sys_Ly_sigma_min_q);
fprintf("Minimum Sigma Value of (1+L_y) for q in db: %d", sys_Ly_sigma_min_db);
```

Minimum Sigma Value of (1+L\_y) for q in db: 2.983842e-01

## 1.8) Classic Stability Margins at the Plant Input $L_u$

```
sys_Lu_margins = allmargin(sys_Lu)
```

```
sys_Lu_margins = struct with fields:
    GainMargin: [0.2481 10.9679]
    GMFrequency: [4.5341 Inf]
    PhaseMargin: 71.0668
    PMFrequency: 32.9302
    DelayMargin: 0.0377
```

DMFrequency: 32.9302  
Stable: 1

```
classical_GM = sys_Lu_margins.GainMargin;
classical_GM_db = 20*log10(classical_GM);
classical_pm = sys_Lu_margins.PhaseMargin;

% Computing the Singular Values stability margins from Part 1.5, (I+L_u)
pm = 2*asin(sys_Lu_sigma_min/2)*180/pi;
GM_l = 20*log10(1/(1+sys_Lu_sigma_min));
GM_u = 20*log10(1/(1-sys_Lu_sigma_min));
GM_IL_u = [GM_l GM_u];
pm_IL_u = [-pm pm];

% Computing the Singular Values stability margins from Part 1.6, (I+L_u^(-1))
pm = 2*asin(sys_Lu_sigma1_min/2)*180/pi;
GM_l = 20*log10((1-sys_Lu_sigma1_min));
GM_u = 20*log10((1+sys_Lu_sigma1_min));
GM_IL_u_inv = [GM_l GM_u];
pm_IL_u_inv = [-pm pm];

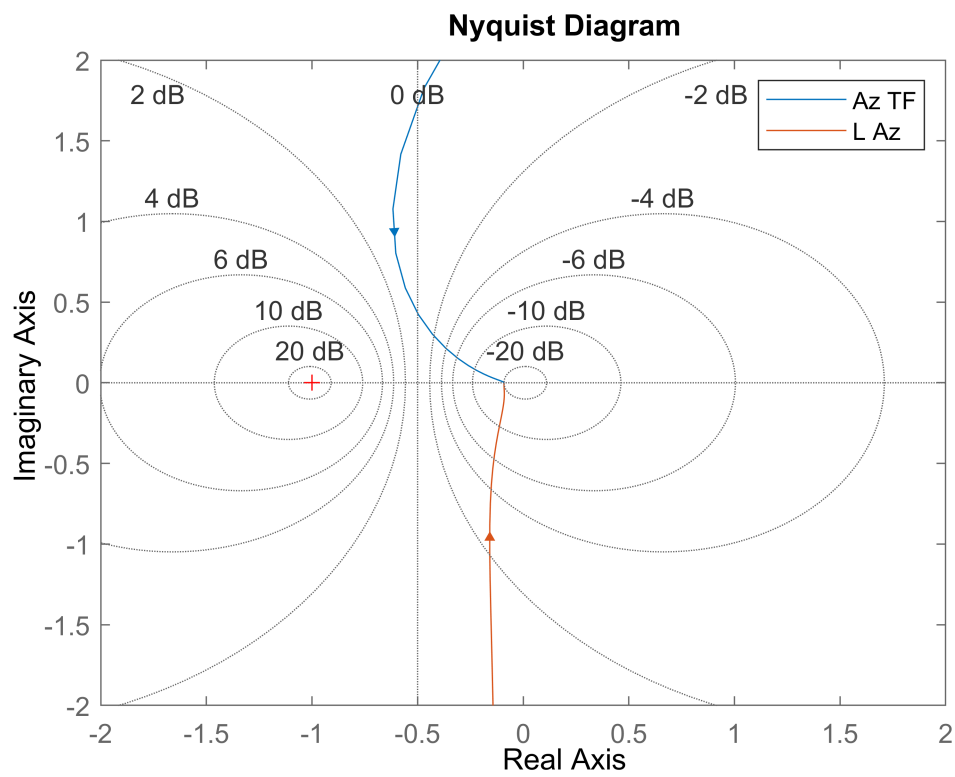
GM_sigma = [min(GM_IL_u_inv(1),GM_IL_u(1)),max(GM_IL_u_inv(2),GM_IL_u(2))];
PM_sigma = [min(pm_IL_u_inv(1),pm_IL_u(1)),max(pm_IL_u_inv(2),pm_IL_u(2))];

text = ['Classical Gain Margin: ',num2str(classical_GM_db(1)),' ',num2str(classical_GM_db(2)),' dB',
newline 'Classical Phase Margin: ',num2str(-classical_pm),' ',num2str(classical_pm),' degrees',
newline 'Singular Values Gain Margin: ',num2str(GM_sigma(1)),' ',num2str(GM_sigma(2)),' dB',
newline 'Singular Values Phase Margin: ',num2str(PM_sigma(1)),' ',num2str(PM_sigma(2)),' degrees'];
disp(text);
```

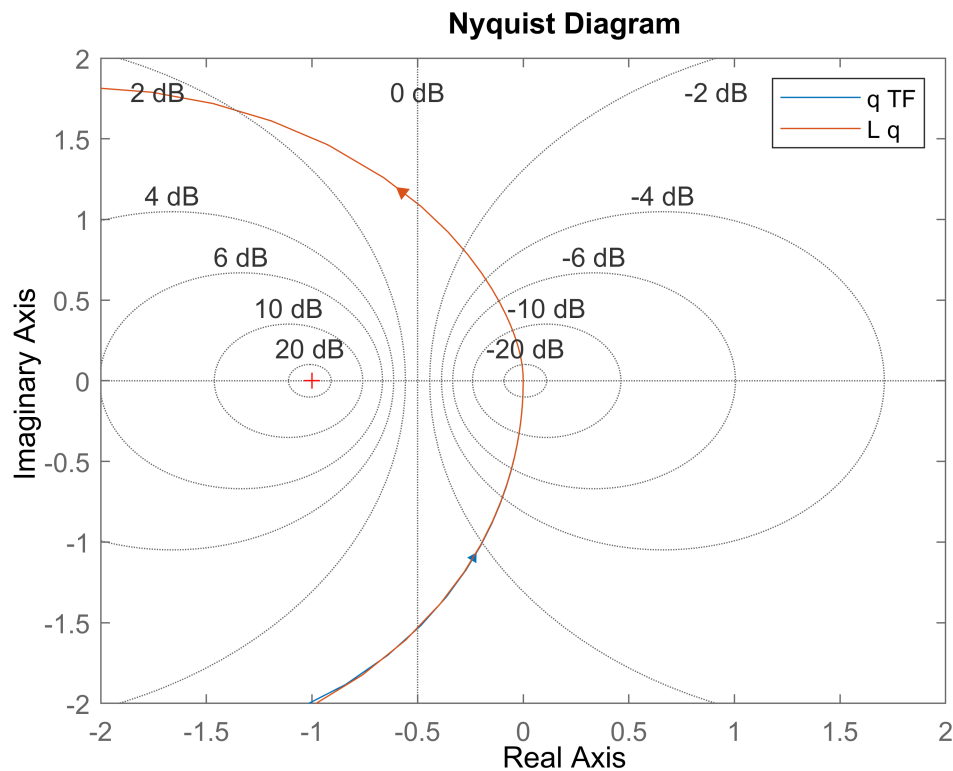
```
Classical Gain Margin: [-12.1067,20.8025] dB
Classical Phase Margin: [-71.0668,71.0668] degrees
Singular Values Gain Margin: [-11.6526,20.8344] dB
Singular Values Phase Margin: [-54.0758,54.0758] degrees
```

## 1.9) Classic Stability Margins at the Plant Output $L_y$

```
% Plotting Nyquist Plots to visualize margins
n = nyquistplot(sys_Ly(1,1),Ly_Az_Open(1,1));
setoptions(n,'ShowfullContour','off'), grid on
legend('Az TF', 'L Az')
xlim([-2 2])
ylim([-2 2])
```



```
n = nyquistplot(sys_Ly(2,2),Ly_q_Open(2,2));
setoptions(n,'ShowfullContour','off'), grid on
legend('q TF', 'L q')
xlim([-2 2])
ylim([-2 2])
```



```
Ly_m_Az_tf = allmargin(sys_Ly(1,1));
Gm_classical_Az_tf = 20*log10(Ly_m_Az_tf.GainMargin);
Pm_classical_Az_tf = Ly_m_Az_tf.PhaseMargin;
Ly_m_q_tf = allmargin(sys_Ly(2,2));
Gm_classical_q_tf = 20*log10(Ly_m_q_tf.GainMargin);
Pm_classical_q_tf = Ly_m_q_tf.PhaseMargin;
text = ['Classical Gain Margin of Az TF: [-inf,',num2str(Gm_classical_Az_tf(1)),'] dB',...
newline 'Classical Phase Margin of Az TF: ',num2str(Pm_classical_Az_tf),' degrees',...
newline 'Classical Gain Margin of q TF:[',num2str(Gm_classical_q_tf(1)),', inf] dB',...
newline 'Classical Phase Margin of q TF: ',num2str(Pm_classical_q_tf),' degrees'];
disp(text);
```

```
Classical Gain Margin of Az TF: [-inf,20.8025] dB
Classical Phase Margin of Az TF: -52.9458 degrees
Classical Gain Margin of q TF: [-13.1144, inf] dB
Classical Phase Margin of q TF: 79.3179 degrees
```

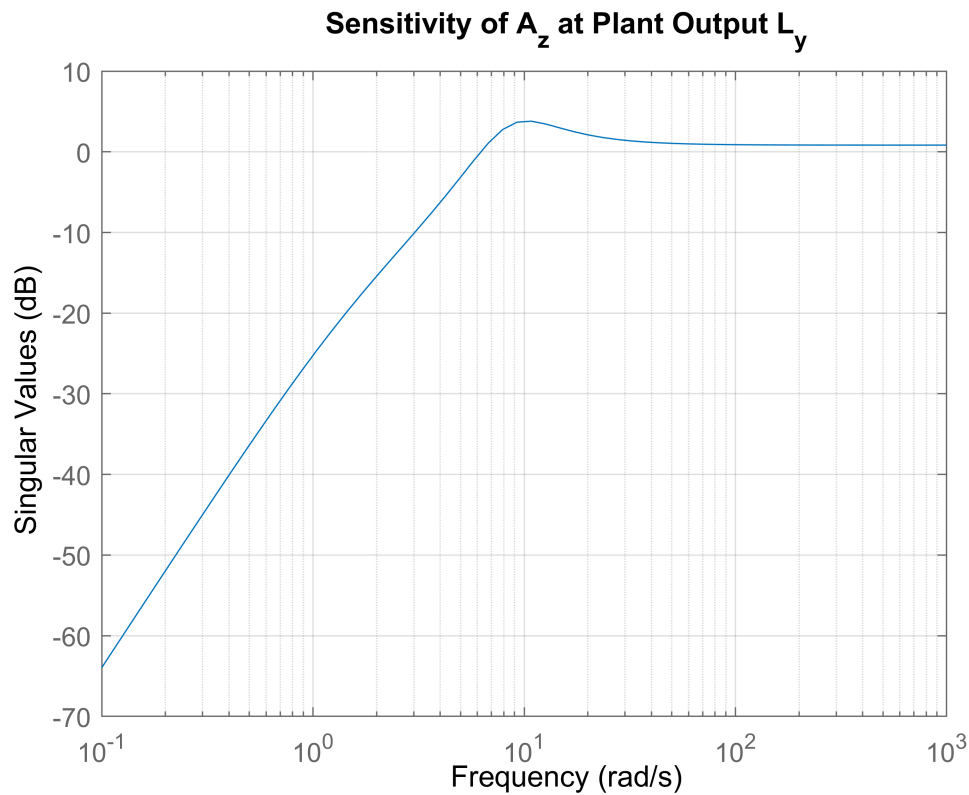
The margins at the plant output are much wider than the margins found in part 1.8 especially the gain margins, this can be interpreted that large amounts of gain inserted at the plant output can be adjusted more without causing instability than inserting those gains at the plant input.

## 1.10) Plotting the Complementary Sensitivity T and Sensitivity S for the Acceleration Loop at Plant Output

```

Sensitivity = 1/(1+sys_Ly(1,1));
sigma(Sensitivity,{.1,1000})
Sensitivity_max = max(sigma(Sensitivity,{.1,1000}));
grid on
title("Sensitivity of A_z at Plant Output L_y")

```



```

Sensitivity_max_db = 20*log10(Sensitivity_max);
text = ['Peak of The Sensitivity S is: ',num2str(Sensitivity_max),' or ',num2str(Sensitivity_max_db)];
disp(text);

```

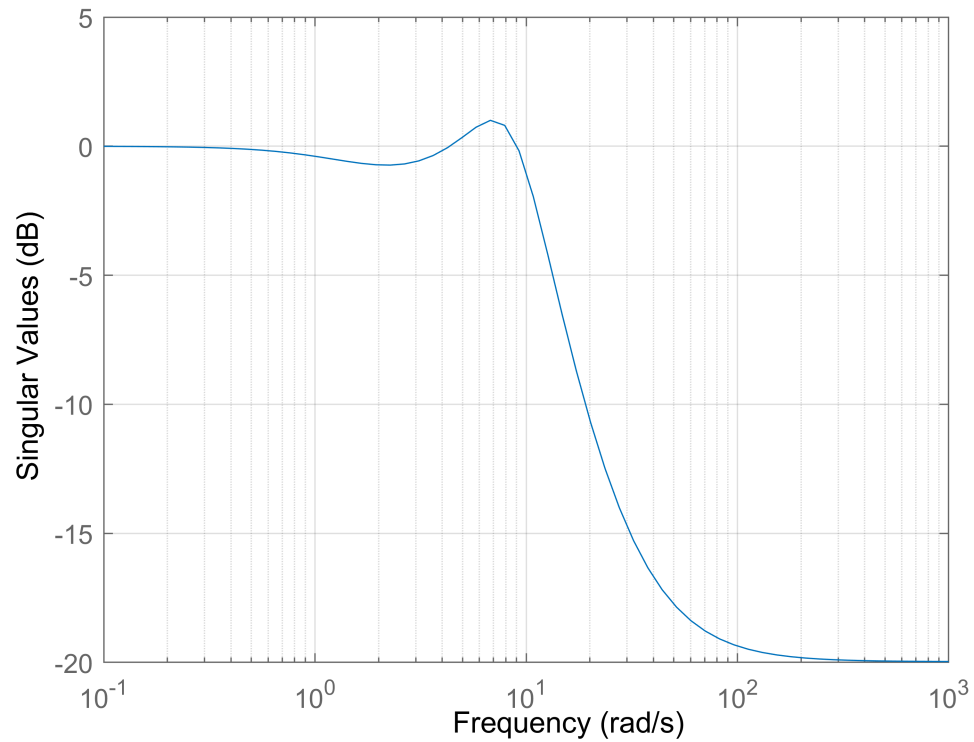
Peak of The Sensitivity S is: 1.5482 or 3.7966 in db

```

Compl_Sensitivity = ((1+sys_Ly(1,1))^-1)*sys_Ly(1,1);
sigma(Compl_Sensitivity,{.1,1000})
Compl_Sensitivity_max = max(sigma(Compl_Sensitivity,{.1,1000}));
grid on
title("Complementary Sensitivity of A_z at Plant Output L_y")

```

### Complementary Sensitivity of $A_z$ at Plant Output $L_y$



```
Compl_Sensitivity_max_db = 20*log10(Compl_Sensitivity_max);  
text = ['Peak of The Complimentary Sensitivity T is: ', num2str(Compl_Sensitivity_max), ' or ', num2str(Compl_Sensitivity_max_db), ' dB'];  
disp(text);
```

Peak of The Complimentary Sensitivity T is: 1.1223 or 1.0023 in db



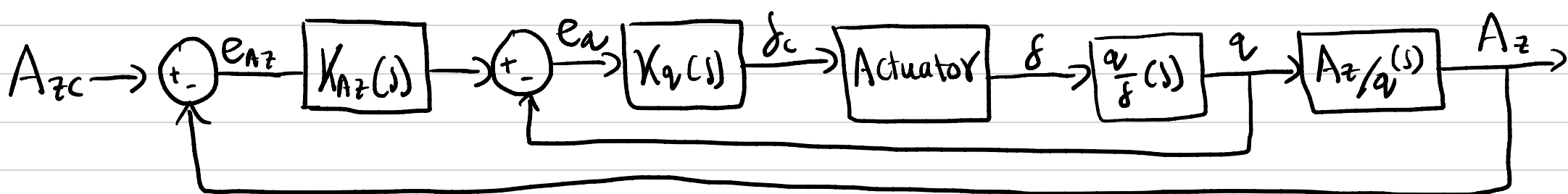
a) Repeat problem 1) and include in the aircraft dynamics a second order actuator

$$\frac{\delta}{\delta_c} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

where  $\omega_n = 150$  (rps) and  $\zeta = 0.6$ .

2.1) Find the Plant Model:

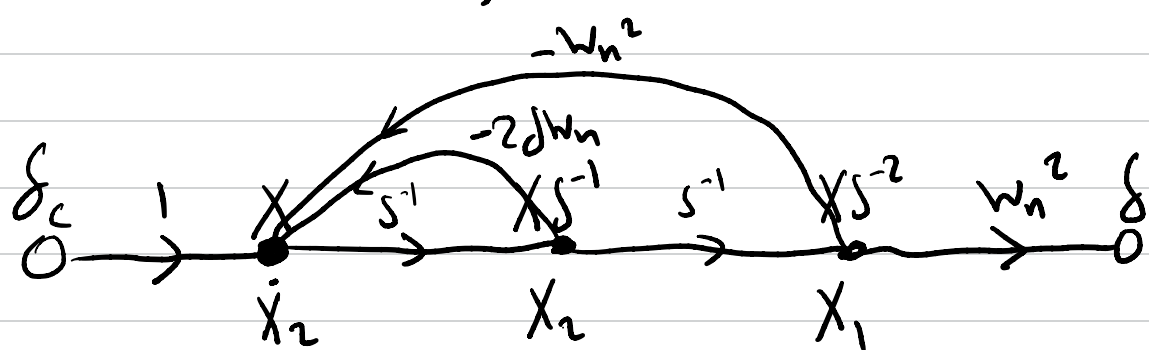
Block Diagram:



Signal Flow Graph For Actuator:

$$\frac{\delta}{\delta_c} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \cdot \frac{X(s)s^{-2}}{X(s)s^{-2}} = \frac{X(s)s^{-2}\omega_n^2}{X(s)s^{-2}\omega_n^2 + X(s)2\zeta\omega_n s^{-1} + X(s)}$$

$$\delta = X(s)s^{-2}\omega_n^2, \quad X(s) = \delta_c - X(s)s^{-2}\omega_n^2 - X(s)s^{-1}2\zeta\omega_n$$



$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\zeta\omega_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \delta_c$$

$$\delta = \begin{bmatrix} \omega_n^2 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 & 0 \end{bmatrix} \delta_c$$

$$\begin{bmatrix} \dot{\alpha} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} Z_{\alpha}/V & 1 \\ M_{\alpha} & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ q \end{bmatrix} + \begin{bmatrix} Z_{\delta}/V \\ M_{\delta} \end{bmatrix} \delta$$

$$\begin{bmatrix} Az \\ q \end{bmatrix} = \begin{bmatrix} Z_{\alpha} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha \\ q \end{bmatrix} + \begin{bmatrix} Z_{\delta} \\ 0 \end{bmatrix} \delta$$

Plant Model with Actuator:

$$\begin{bmatrix} \dot{x} \\ \dot{x}_a \end{bmatrix} = \begin{bmatrix} A_p & B_p C_a \\ 0 & A_a \end{bmatrix} \begin{bmatrix} x \\ x_a \end{bmatrix} + \begin{bmatrix} 0 \\ B_a \end{bmatrix} \delta_c$$

$$y = \begin{bmatrix} C_p & D_p C_a \end{bmatrix} \begin{bmatrix} x \\ x_a \end{bmatrix} + 0 \delta_c$$

Complete Closed-loop Model:

- Same as before but:

$$A_p = \begin{bmatrix} A_p & B_p C_a \\ 0 & A_a \end{bmatrix}, \quad B_p = \begin{bmatrix} 0 \\ B_a \end{bmatrix}$$

$$C_p = \begin{bmatrix} C_p & D_p C_a \end{bmatrix}, \quad D_p = 0$$

# Aircraft Pitch Controller With Second Order Actuator

```
% Initializing symbolic Variables
syms alpha_dot q_dot Z_alpha M_alpha V alpha q Z_delta M_delta delta_e ...
a_z a_q K_q K_a alpha_dot_c q_dot_c alpha_c q_c w_n zeta delta_c x_dot_1 ...
x_dot_2 x_1 x_2 delta delta_c
```

## Building the Actuator State Space Model

```
A_a = [0 1; -w_n^2 -2*zeta*w_n];
B_a = [0; 1];
C_a = [w_n^2 0];
D_a = [0 0];
X_dot_a = [x_dot_1; x_dot_2];
X_a = [x_1; x_2];
y_a = delta;
u_a = delta_c;
% Displaying the Plant State Space Model
s = ["X_dot_a = A_a*X_a + B_a*u_a"; "y_a = C_a*X_a + D_a*u_a"];
displayFormula(["'Actuator State Space Model:'"; s]);
```

Actuator State Space Model:

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -w_n^2 & -2 w_n \zeta \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \delta_c$$

$$\delta = (w_n^2 \ 0) \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + (0 \ 0) \delta_c$$

## Building the Controller State Space Model

```
A_c = [0 0; K_a*a_z 0];
B_c1 = [-1 0; -K_a -1];
B_c2 = [1; K_a];
C_c = [K_q*K_a*a_z K_q*a_q];
D_c1 = [-K_q*K_a -K_q];
D_c2 = [K_q*K_a];
X_dot_c = [alpha_dot_c; q_dot_c];
X_c = [alpha_c; q_c];
s = ["X_dot_c = A_c*X_c + B_c1*y + B_c2*r"; "delta_e = C_c*X_c + D_c1*y + D_c2*r"];
displayFormula(["'Controller State Space Model:'"; s]);
```

Controller State Space Model:

$$\begin{pmatrix} \dot{\alpha}_c \\ \dot{q}_c \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ K_a a_z & 0 \end{pmatrix} \begin{pmatrix} \alpha_c \\ q_c \end{pmatrix} + \begin{pmatrix} -1 & 0 \\ -K_a & -1 \end{pmatrix} \begin{pmatrix} a_z \\ q \end{pmatrix} + \begin{pmatrix} 1 \\ K_a \end{pmatrix} r$$

$$\delta_e = (K_a K_q a_z \ K_q a_q) \begin{pmatrix} \alpha_c \\ q_c \end{pmatrix} + (-K_a K_q \ -K_q) \begin{pmatrix} a_z \\ q \end{pmatrix} + K_a K_q r$$

## 2.1.i) Building the Plant State Space Model To Include Actuator

```
% Plant matrices before actuator
A_pp = [Z_alpha/V 1;M_alpha 0];
B_pp = [Z_delta/V;M_delta];
C_pp = [Z_alpha 0;0 1];
D_pp = [Z_delta;0];

% Plant matrices after actuator
A_p = [A_pp B_pp*C_a;zeros(2) A_a];
B_p = [0;0;B_a];
C_p = [C_pp D_pp*C_a];
D_p = [0;0];

X_dot = [alpha_dot;q_dot;X_dot_a];
X = [alpha;q;X_a];
y = [a_z;q];
u = u_a;
% Displaying the Plant State Space Model
s = ["X_dot = A_p*X + B_p*u";"y = C_p*X + D_p*u"];
displayFormula(["'Plant State Space Model:'";s]);
```

Plant State Space Model:

$$\begin{pmatrix} \dot{\alpha} \\ \dot{q} \\ \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} \frac{Z_\alpha}{V} & 1 & \frac{Z_\delta w_n^2}{V} & 0 \\ M_\alpha & 0 & M_\delta w_n^2 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -w_n^2 & -2 w_n \zeta \end{pmatrix} \begin{pmatrix} \alpha \\ q \\ x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \delta_c$$

$$\begin{pmatrix} a_z \\ q \end{pmatrix} = \begin{pmatrix} Z_\alpha & 0 & Z_\delta w_n^2 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ q \\ x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix} \delta_c$$

## 2.1.ii) Building the Closed Loop State Space Model

```
% I = eye(size(D_c1*D_p));
% z = (I-D_c1*D_p);
A_cl = [A_p+(B_p*D_c1*C_p) B_p*C_c;...
B_c1*(eye(size(D_p*D_c1))+D_p*D_c1)*C_p A_c+B_c1*D_p*C_c];
B_cl = [B_p*D_c2;B_c2+B_c1*D_p*D_c2];
C_cl = [(eye(size(D_p*D_c1))+D_p*D_c1)*C_p D_p*C_c];
D_cl = [D_p*D_c2];
X_dot_cl = [X_dot;X_dot_c];
X_cl = [X;X_c];
s = ["X_dot_cl = A_cl*X_cl + B_cl*r";"y= C_cl*X_cl + D_cl*r"];
displayFormula(["'Complete Closed Loop State Space Model:'";s]);
```

Complete Closed Loop State Space Model:

$$\begin{pmatrix} \dot{\alpha} \\ \dot{q} \\ \dot{x}_1 \\ \dot{x}_2 \\ \dot{\alpha}_c \\ \dot{q}_c \end{pmatrix} = \begin{pmatrix} \frac{Z_\alpha}{V} & 1 & \frac{Z_\delta w_n^2}{V} & 0 & 0 & 0 \\ M_\alpha & 0 & M_\delta w_n^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ -K_a K_q Z_\alpha & -K_q & -w_n^2 - K_a K_q Z_\delta w_n^2 & -2 w_n \zeta & K_a K_q a_z & K_q a_q \\ -Z_\alpha & 0 & -Z_\delta w_n^2 & 0 & 0 & 0 \\ -K_a Z_\alpha & -1 & -K_a Z_\delta w_n^2 & 0 & K_a a_z & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ q \\ x_1 \\ x_2 \\ \alpha_c \\ q_c \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ K_a K_q \\ 1 \\ K_a \end{pmatrix} r$$

$$\begin{pmatrix} a_z \\ q \end{pmatrix} = \begin{pmatrix} Z_\alpha & 0 & Z_\delta w_n^2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ q \\ x_1 \\ x_2 \\ \alpha_c \\ q_c \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix} r$$

## 2.1.ii) Subbing in the Numerical Values to the Closed Loop Model

```
% V = 886.78; % Feet per second
% Z_alpha = -1.3046*V;
% Z_delta = -0.2142*V;
% M_alpha = 47.7109;
% M_delta = -104.8346;
% K_a = -0.0015;
% K_q = -0.32;
% a_z = 2.0;
% a_q = 6.0;
% w_n = 150; % rps
% zeta = 0.6;
variables = [V;Z_alpha;Z_delta;M_alpha;M_delta;K_a;K_q;a_z;a_q;w_n;zeta];
variable_values = [886.78;-1.3046*886.78;-0.2142*886.78;47.7109;-104.8346;-0.0015;-0.32;2.0;6.0;150;0.6];
A_cl = double(subs(A_cl,variables,variable_values));
B_cl = double(subs(B_cl,variables,variable_values));
C_cl = double(subs(C_cl,variables,variable_values));
D_cl = double(subs(D_cl,variables,variable_values));
A_cl0 = round(vpa(A_cl,2));
B_cl0 = round(vpa(B_cl,2));
C_cl0 = round(vpa(C_cl,2));
D_cl0 = round(vpa(D_cl,2));

s = ["X_dot_cl = A_cl0*X_cl + B_cl0*r";"y= C_cl0*X_cl + D_cl0*r"];
displayFormula(['Complete Closed Loop State Space Model:'];s]);
```

Complete Closed Loop State Space Model:

$$\begin{pmatrix} \dot{\alpha} \\ \dot{q} \\ \dot{x}_1 \\ \dot{x}_2 \\ \dot{\alpha}_c \\ \dot{q}_c \end{pmatrix} = \begin{pmatrix} -1 & 1 & -4820 & 0 & 0 & 0 \\ 48 & 0 & -2358778 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & -20449 & -180 & 0 & -2 \\ 1157 & 0 & 4273836 & 0 & 0 & 0 \\ -2 & -1 & -6411 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ q \\ x_1 \\ x_2 \\ \alpha_c \\ q_c \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} r$$

$$\begin{pmatrix} a_z \\ q \end{pmatrix} = \begin{pmatrix} -1157 & 0 & -4273836 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ q \\ x_1 \\ x_2 \\ \alpha_c \\ q_c \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix} r$$

```
A_cl_eig = double(eig(A_cl));
disp(["Eigenvalues of Closed Loop System: "; num2str(A_cl_eig)]);
```

```
"Eigenvalues of Closed Loop System: "
"-64.56458+96.46013i"
"-64.56458-96.46013i"
"-42.96583+0i"
"-3.691956+1.534035i"
"-3.691956-1.534035i"
"-1.825691+0i"
```

## 2.1.iii) Loop Transfer Function at the Plant Input L\_u

```
A_Lu = [A_p 0.*B_p*C_c;B_c1*C_p A_c];
B_Lu = [B_p;B_c1*D_p];
C_Lu = -[D_c1*C_p C_c];
D_Lu = -[D_c1*D_p];
s = ["X_dot_cl = A_Lu*X_cl + B_Lu*u_in";"u_out= C_Lu*X_cl + D_Lu*u_in"];
displayFormula(["'Loop Opened at Plant Input:'";s]);
```

Loop Opened at Plant Input:

$$\begin{pmatrix} \dot{\alpha} \\ \dot{q} \\ \dot{x}_1 \\ \dot{x}_2 \\ \dot{\alpha}_c \\ \dot{q}_c \end{pmatrix} = \begin{pmatrix} \frac{Z_\alpha}{V} & 1 & \frac{Z_\delta w_n^2}{V} & 0 & 0 & 0 \\ M_\alpha & 0 & M_\delta w_n^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -w_n^2 & -2 w_n \zeta & 0 & 0 \\ -Z_\alpha & 0 & -Z_\delta w_n^2 & 0 & 0 & 0 \\ -K_a Z_\alpha & -1 & -K_a Z_\delta w_n^2 & 0 & K_a a_z & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ q \\ x_1 \\ x_2 \\ \alpha_c \\ q_c \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} u_{in}$$

$$u_{\text{out}} = \begin{pmatrix} K_a K_q Z_\alpha & K_q & K_a K_q Z_\delta w_n^2 & 0 & -K_a K_q a_z & -K_q a_q \end{pmatrix} \begin{pmatrix} \alpha \\ q \\ x_1 \\ x_2 \\ \alpha_c \\ q_c \end{pmatrix} + 0 u_{\text{in}}$$

```
A_Lu = double(subs(A_Lu,variables,variable_values));
B_Lu = double(subs(B_Lu,variables,variable_values));
C_Lu = double(subs(C_Lu,variables,variable_values));
D_Lu = double(subs(D_Lu,variables,variable_values));
A_Lu0 = round(vpa(A_Lu,2));
B_Lu0 = round(vpa(B_Lu,2));
C_Lu0 = round(vpa(C_Lu,2));
D_Lu0 = round(vpa(D_Lu,2));
s = ["X_dot_cl = A_Lu0*X_cl + B_Lu0*u_in"; "u_out= C_Lu0*X_cl + D_Lu0*u_in"];
displayFormula(["'Loop Opened at Plant Input:'";s]);
```

Loop Opened at Plant Input:

$$\begin{pmatrix} \dot{\alpha} \\ \dot{q} \\ \dot{x}_1 \\ \dot{x}_2 \\ \dot{\alpha}_c \\ \dot{q}_c \end{pmatrix} = \begin{pmatrix} -1 & 1 & -4820 & 0 & 0 & 0 \\ 48 & 0 & -2358778 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -22500 & -180 & 0 & 0 \\ 1157 & 0 & 4273836 & 0 & 0 & 0 \\ -2 & -1 & -6411 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ q \\ x_1 \\ x_2 \\ \alpha_c \\ q_c \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} u_{\text{in}}$$

$$u_{\text{out}} = \begin{pmatrix} -1 & 0 & -2051 & 0 & 0 & 2 \end{pmatrix} \begin{pmatrix} \alpha \\ q \\ x_1 \\ x_2 \\ \alpha_c \\ q_c \end{pmatrix} + 0 u_{\text{in}}$$

```
sys_Lu = ss(A_Lu,B_Lu,C_Lu,D_Lu);
[b,a] = ss2tf(A_Lu,B_Lu,C_Lu,D_Lu);
sys_Lu_TF = tf(b,a)
```

sys\_Lu\_TF =

$$\frac{-2051 s^4 + 7.384e05 s^3 + 6.97e06 s^2 + 1.761e07 s + 1.689e07}{s^6 + 181.3 s^5 + 2.269e04 s^4 + 2.077e04 s^3 - 1.073e06 s^2}$$

Continuous-time transfer function.

## 2.1.iv) Loop Transfer Function at the Plant Output L\_y

```

A_Ly = [A_p B_p*C_c;0.*B_c1*C_p A_c];
B_Ly = [B_p*D_c1;B_c1];
C_Ly = -[C_p D_p*C_c];
D_Ly = -[D_p*D_c1];
s = ["X_dot_cl = A_Ly*X_cl + B_Ly*y_in";"y_out= C_Ly*X_cl + D_Ly*y_in"];
displayFormula(['''Loop Opened at Plant Output:''';s]);

```

Loop Opened at Plant Output:

$$\begin{pmatrix} \dot{\alpha} \\ \dot{q} \\ \dot{x}_1 \\ \dot{x}_2 \\ \dot{\alpha}_c \\ \dot{q}_c \end{pmatrix} = \begin{pmatrix} \frac{Z_\alpha}{V} & 1 & \frac{Z_\delta w_n^2}{V} & 0 & 0 & 0 \\ M_\alpha & 0 & M_\delta w_n^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -w_n^2 & -2 w_n \zeta & K_a K_q a_z & K_q a_q \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & K_a a_z & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ q \\ x_1 \\ x_2 \\ \alpha_c \\ q_c \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -K_a K_q & -K_q \\ -1 & 0 \\ -K_a & -1 \end{pmatrix} y_{in}$$

$$y_{out} = \begin{pmatrix} -Z_\alpha & 0 & -Z_\delta w_n^2 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ q \\ x_1 \\ x_2 \\ \alpha_c \\ q_c \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} y_{in}$$

```

A_Ly = double(subs(A_Ly,variables,variable_values));
B_Ly = double(subs(B_Ly,variables,variable_values));
C_Ly = double(subs(C_Ly,variables,variable_values));
D_Ly = double(subs(D_Ly,variables,variable_values));
sys_Ly = ss(A_Ly,B_Ly,C_Ly,D_Ly);
A_Ly0 = round(vpa(A_Ly,2));
B_Ly0 = round(vpa(B_Ly,2));
C_Ly0 = round(vpa(C_Ly,2));
D_Ly0 = round(vpa(D_Ly,2));
s = ["X_dot_cl = A_Ly0*X_cl + B_Ly0*y_in";"y_out= C_Ly0*X_cl + D_Ly0*y_in"];
displayFormula(['''Loop Opened at Plant Output:''';s]);

```

Loop Opened at Plant Output:

$$\begin{pmatrix} \dot{\alpha} \\ \dot{q} \\ \dot{x}_1 \\ \dot{x}_2 \\ \dot{\alpha}_c \\ \dot{q}_c \end{pmatrix} = \begin{pmatrix} -1 & 1 & -4820 & 0 & 0 & 0 \\ 48 & 0 & -2358778 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -22500 & -180 & 0 & -2 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ q \\ x_1 \\ x_2 \\ \alpha_c \\ q_c \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -1 & 0 \\ 0 & -1 \end{pmatrix} y_{in}$$

$$y_{out} = \begin{pmatrix} 1157 & 0 & 4273836 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ q \\ x_1 \\ x_2 \\ \alpha_c \\ q_c \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} y_{in}$$

```
[b,a] = ss2tf(A_Lu,B_Lu,C_Lu,D_Lu);
sys_Ly_TF = tf(b,a)
```

```
sys_Ly_TF =
```

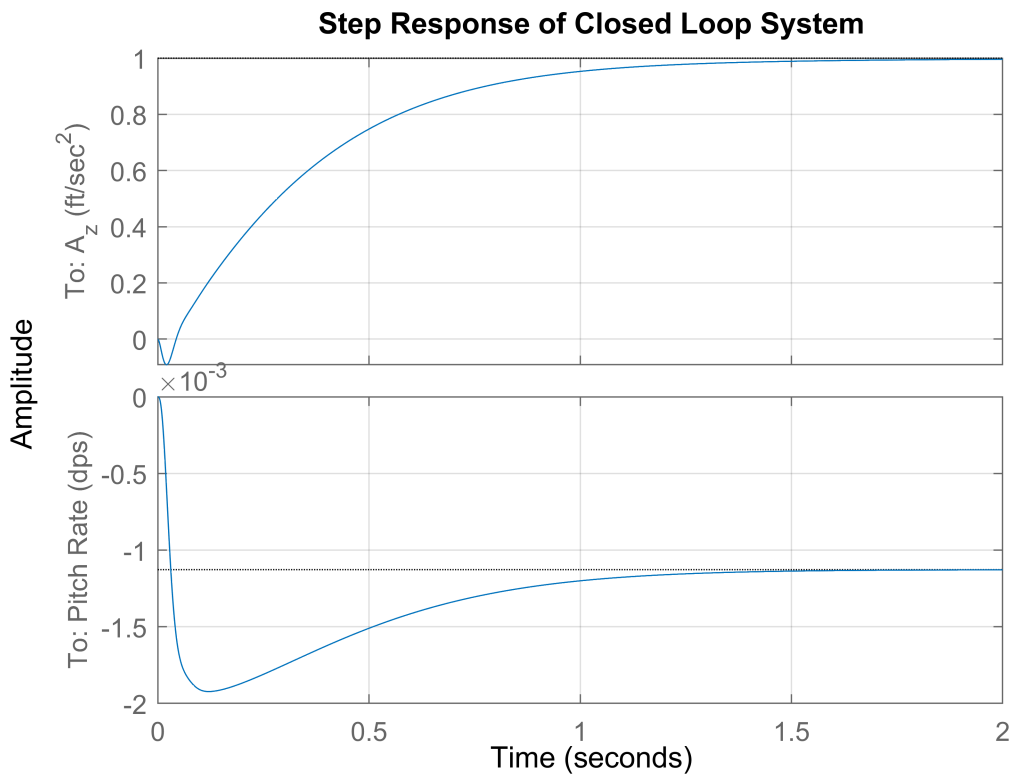
```
-2051 s^4 + 7.384e05 s^3 + 6.97e06 s^2 + 1.761e07 s + 1.689e07
-----
s^6 + 181.3 s^5 + 2.269e04 s^4 + 2.077e04 s^3 - 1.073e06 s^2
```

Continuous-time transfer function.

## 2.2) Computing the Closed Loop System Step Response, Rise Time & Settling Time

```
sys_cl = ss(A_cl,B_cl,C_cl,D_cl);
sys_cl(1,1).OutputName = 'A_z (ft/sec^2)';
sys_cl(2,1).OutputName = 'Pitch Rate (dps)';
step(sys_cl)
xlim([0,2])
grid on
title("Step Response of Closed Loop System")
```





```
info = stepinfo(sys_cl, 'RiseTimeThreshold', [0 0.63], 'SettlingTimeThreshold', 0.05)
```

```
info = 2x1 struct
```

...

| Fields | RiseTime | TransientTime | SettlingTime | SettlingMin | SettlingMax |
|--------|----------|---------------|--------------|-------------|-------------|
| 1      | 0.3372   | 0.9557        | 0.9821       | 0.63        | 0.9997      |
| 2      | 0.0227   | 1.066         | 1.066        | -0.0019     | -0.00071456 |

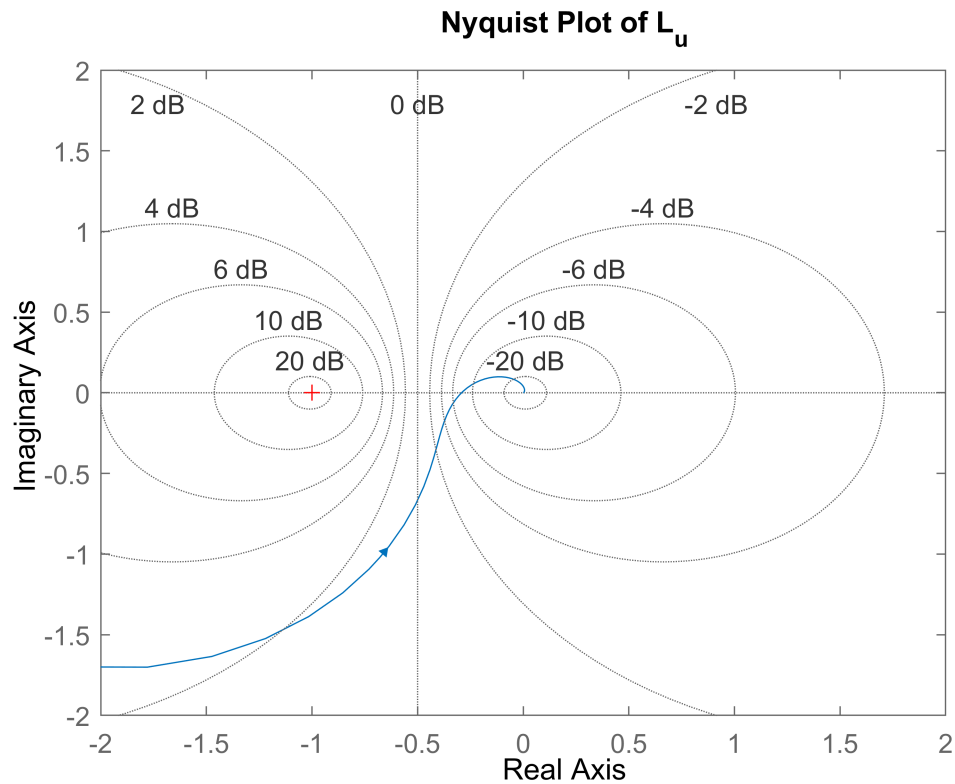
```
text = ['Rise Time of Az (63%): ', num2str(info(1).RiseTime), ' sec', ...
newline 'Settling Time of Az (95%): ', num2str(info(1).SettlingTime), 'sec', ...
newline 'Rise Time of q (63%): ', num2str(info(2).RiseTime), 'sec', ...
newline 'Settling Time of q (95%): ', num2str(info(2).SettlingTime), 'sec'];
disp(text)
```

```
Rise Time of Az (63%): 0.33722 sec,
Settling Time of Az (95%): 0.98209sec
Rise Time of q (63%): 0.022749sec,
Settling Time of q (95%): 1.066sec
```

## 2.3.i) Nyquist Plot of $L_u$

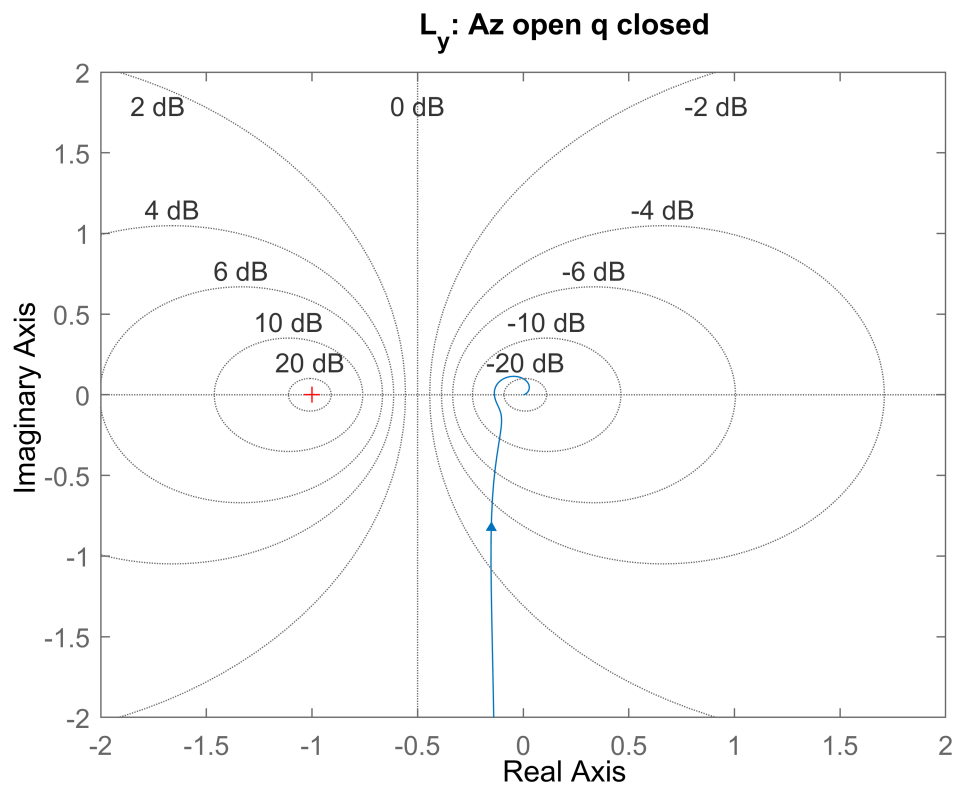
```
h = nyquistplot(sys_Lu);
title("Nyquist Plot of  $L_u$ ")
setoptions(h, 'ShowFullContour', 'off');
grid on
xlim([-2, 2])
```

```
ylim([-2,2])
```



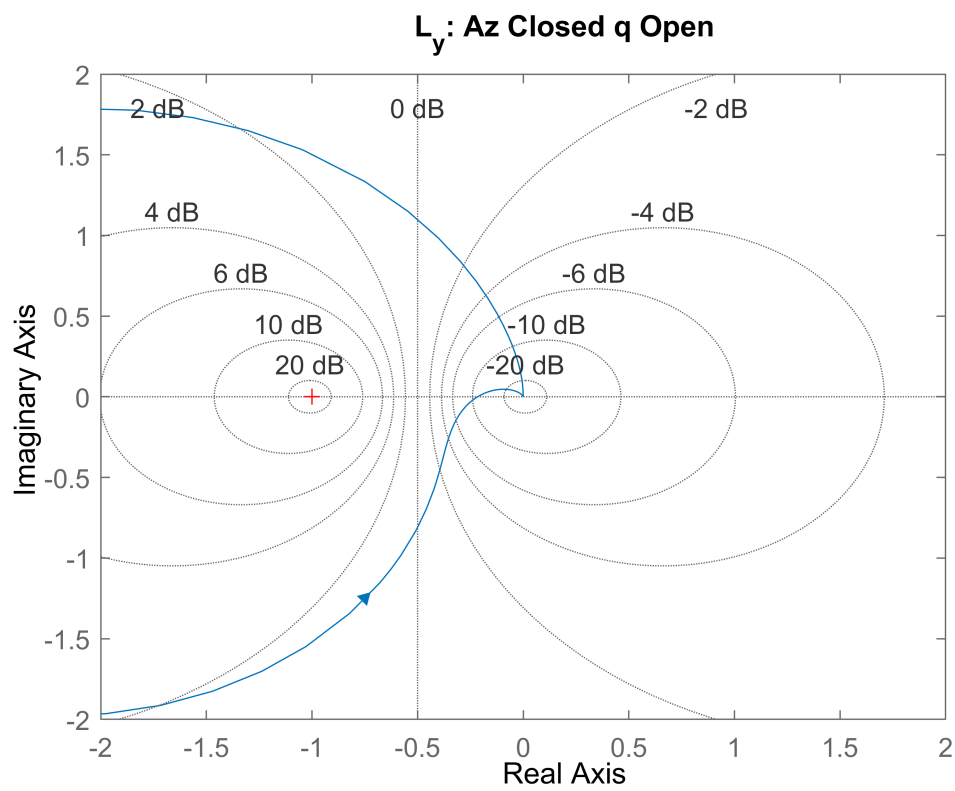
### 2.3.ii) Isolate Outputs: Az open q closed, Nyquist Plot

```
Ly_Az_Open = feedback(sys_Ly,eye(size(sys_Ly,1)-1),2,2);  
h = nyquistplot(Ly_Az_Open(1,1));  
setoptions(h,'ShowFullContour','off');  
grid on  
xlim([-2,2])  
ylim([-2,2])  
title("L_y: Az open q closed")
```



### 2.3.iii) Isolate Outputs: Az Closed q Open, Nyquist Plot

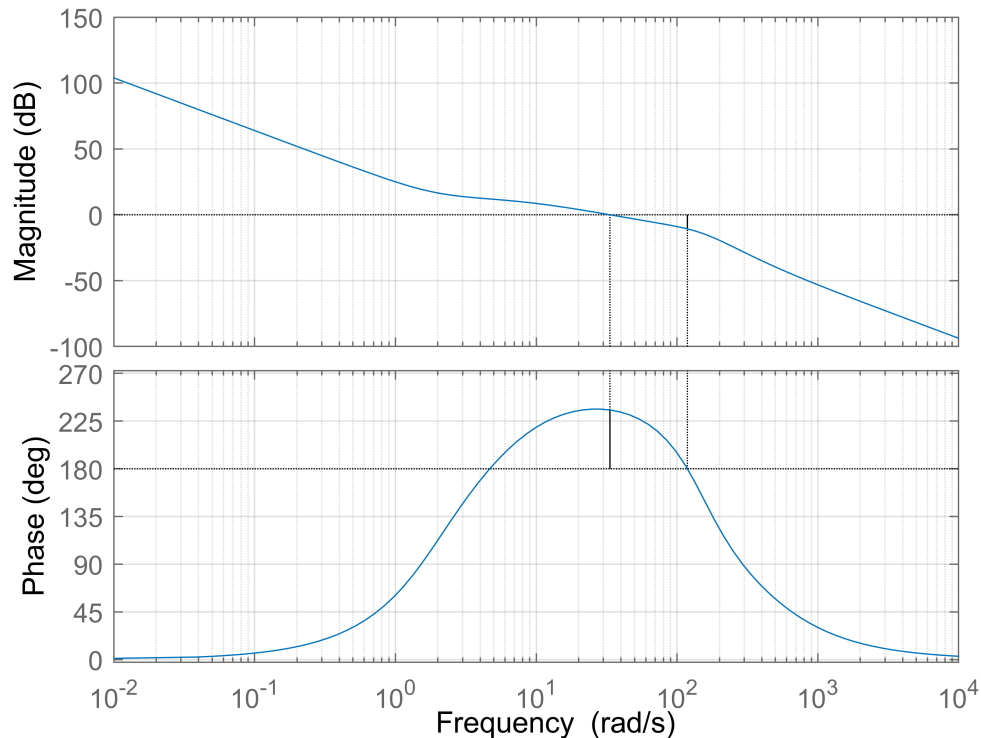
```
Ly_q_Open = feedback(sys_Ly,eye(size(sys_Ly,1)-1),1,1);
h = nyquistplot(Ly_q_Open(2,2));
setoptions(h,'ShowFullContour','off');
grid on
xlim([-2,2])
ylim([-2,2])
title(" $L_y$ : Az Closed q Open")
```



## 2.4) Bode Plot for $L_u$

```
margin(sys_Lu)
grid on
title("Frequency Response at Plant Input ,  $L_u$ ")
```

### Frequency Response at Plant Input , $L_u$



```
lu = allmargin(sys_Lu)
```

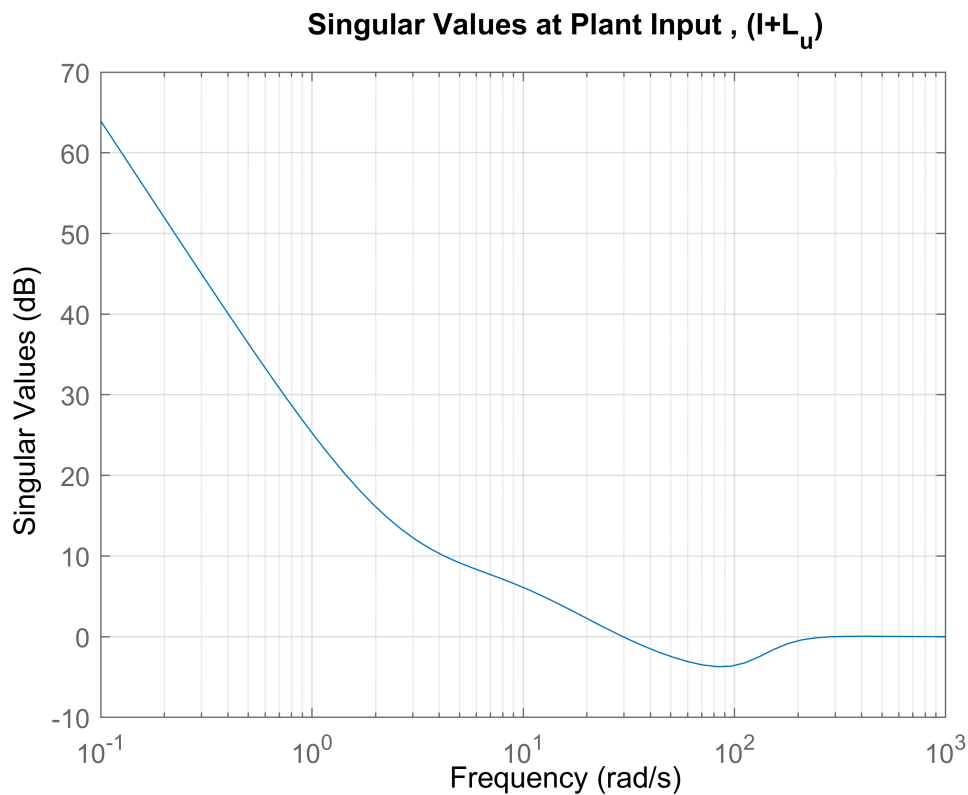
```
lu = struct with fields:
    GainMargin: [0.2514 3.4243]
    GMFrequency: [4.6768 118.4754]
    PhaseMargin: 55.4905
    PMFrequency: 33.3808
    DelayMargin: 0.0290
    DMFrequency: 33.3808
    Stable: 1
```

```
text = ['Gain Margin: ', num2str(20*log10(lu.GainMargin(2))), ' dB',...
        newline 'LGCF @: ', num2str(lu.PMFrequency(1)), ' rad/s',...
        newline 'Phase margin: ' num2str(lu.PhaseMargin), ' degrees',...
        newline 'Phase Crossover @: ', num2str(lu.GMFrequency(1)), ' rad/s'];
disp(text)
```

```
Gain Margin: 10.6915 dB,
LGCF @: 33.3808 rad/s
Phase margin: 55.4905 degrees,
Phase Crossover @: 4.6768 rad/s
```

## 2.5) Plotting the singular values for $(I+L_u)$ & Computing the Minimum Sigma

```
sigma(sys_Lu+1,{.1,1000})
sys_Lu_sigma_min = min(sigma(sys_Lu+1));
grid on
title("Singular Values at Plant Input ,  $(I+L_u)$ ")
```



```
fprintf("Minimum Sigma Value of (1+L_u): %d", sys_Lu_sigma_min);
```

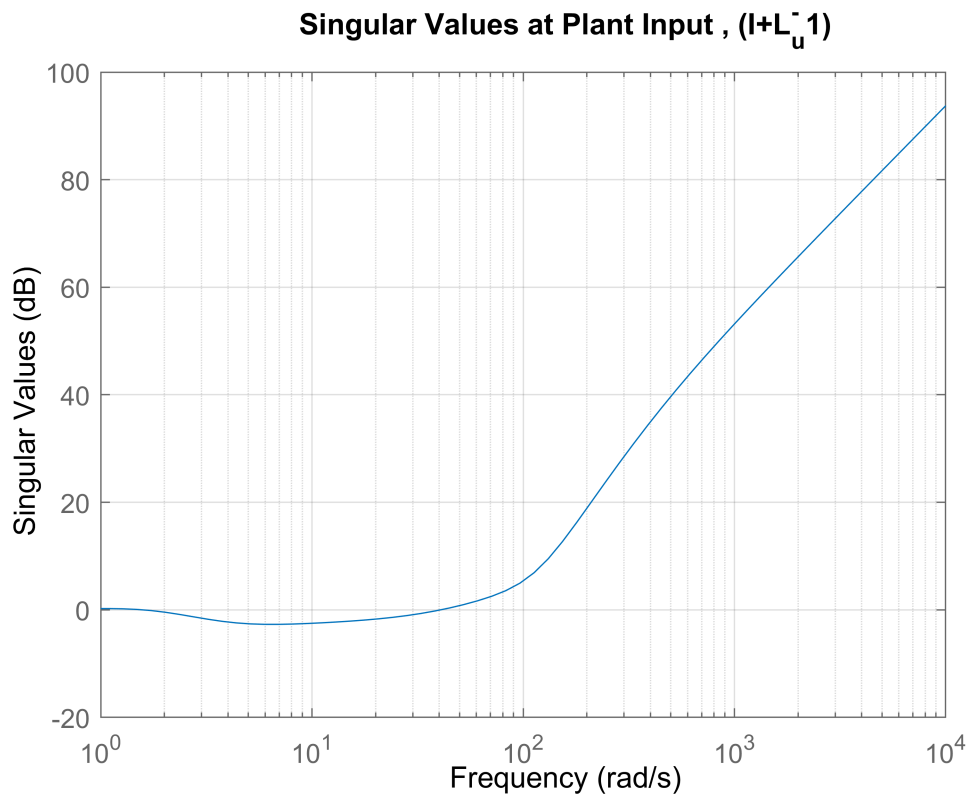
Minimum Sigma Value of (1+L\_u): 6.516401e-01

```
sys_Lu_sigma_min_db = 20*log10(sys_Lu_sigma_min);
fprintf("Minimum Sigma Value of (1+L_u) in db: %d", sys_Lu_sigma_min_db);
```

Minimum Sigma Value of (1+L\_u) in db: -3.719844e+00

## 2.6) Plotting the singular values for $(I+L_u^{-1})$ & Computing the Minimum Sigma

```
sigma(sys_Lu^-1+1)
sys_Lu_sigma1_min = min(sigma(sys_Lu^(-1)+1));
grid on
title("Singular Values at Plant Input ,  $(I+L_u^{-1})$ ")
```



```
fprintf("Minimum Sigma Value of  $(1+L_u^{-1})$ : %d", sys_Lu_sigma1_min);
```

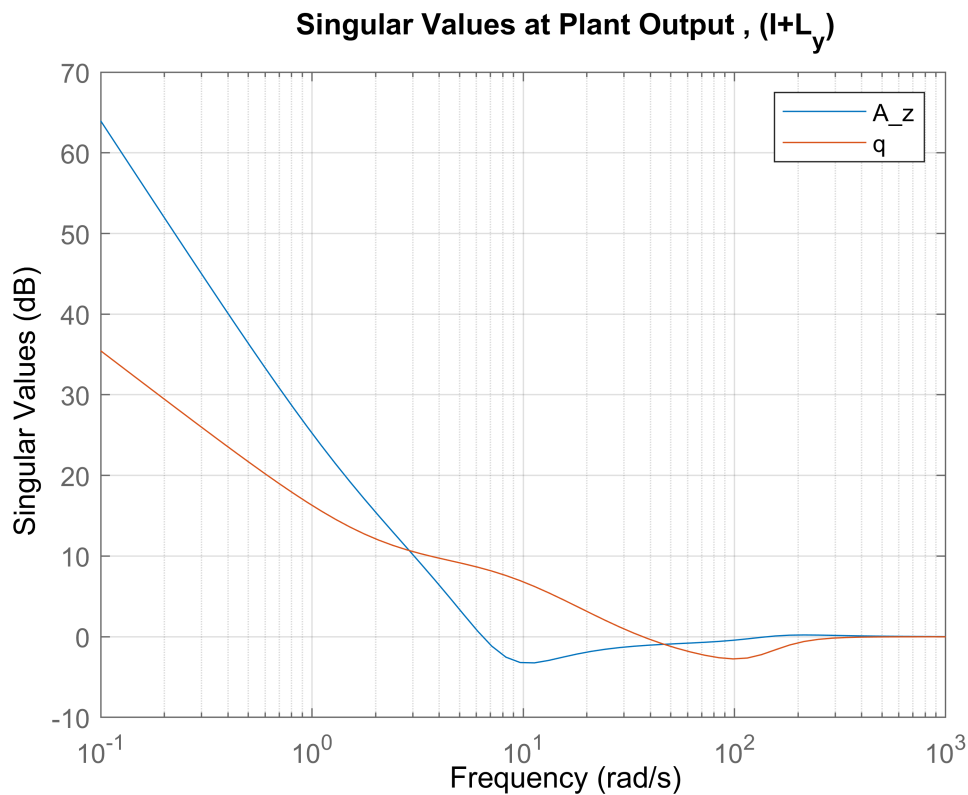
Minimum Sigma Value of  $(1+L_u^{-1})$ : 7.334634e-01

```
sys_Lu_sigma1_min_db = 20*log10(sys_Lu_sigma1_min);
fprintf("Minimum Sigma Value of  $(1+L_u)$  in db: %d", sys_Lu_sigma1_min_db);
```

Minimum Sigma Value of  $(1+L_u)$  in db: -2.692431e+00

## 2.7) Plotting the singular values for $(I+L_y)$

```
sigma(sys_Ly(1,1)+1,sys_Ly(2,2)+1)
sys_Ly_sigma_min_A_z = min(sigma(sys_Ly(1,1)+1));
sys_Ly_sigma_min_q = min(sigma(sys_Ly(2,2)+1));
grid on
title("Singular Values at Plant Output ,  $(I+L_y)$ ")
legend("A_z","q")
```



```
fprintf("Minimum Sigma Value of  $(1+L_y)$  for  $A_z$ : %d", sys_Ly_sigma_min_A_z);
```

Minimum Sigma Value of  $(1+L_y)$  for  $A_z$ : 6.880340e-01

```
sys_Ly_sigma_min_db = 20*log10(sys_Ly_sigma_min_A_z);
fprintf("Minimum Sigma Value of  $(1+L_y)$  for  $A_z$  in db: %d", sys_Ly_sigma_min_db);
```

Minimum Sigma Value of  $(1+L_y)$  for  $A_z$  in db: -3.247802e+00

```
fprintf("Minimum Sigma Value of  $(1+L_y)$  for q: %d", sys_Ly_sigma_min_q);
```

Minimum Sigma Value of  $(1+L_y)$  for q: 7.286759e-01

```
sys_Ly_sigma_min_db = 20*log10(sys_Ly_sigma_min_q);
fprintf("Minimum Sigma Value of  $(1+L_y)$  for q in db: %d", sys_Ly_sigma_min_db);
```

Minimum Sigma Value of  $(1+L_y)$  for q in db: -2.749312e+00

## 2.8) Classic Stability Margins at the Plant Input $L_u$

```
sys_Lu_margins = allmargin(sys_Lu)
```

```
sys_Lu_margins = struct with fields:
    GainMargin: [0.2514 3.4243]
    GMFrequency: [4.6768 118.4754]
    PhaseMargin: 55.4905
    PMFrequency: 33.3808
    DelayMargin: 0.0290
```



DMFrequency: 33.3808  
Stable: 1

```
classical_GM = sys_Lu_margins.GainMargin;
classical_GM_db = 20*log10(classical_GM);
classical_pm = sys_Lu_margins.PhaseMargin;

% Computing the Singular Values stability margins from Part 1.5, (I+L_u)
pm = 2*asin(sys_Lu_sigma_min/2)*180/pi;
GM_l = 20*log10(1/(1+sys_Lu_sigma_min));
GM_u = 20*log10(1/(1-sys_Lu_sigma_min));
GM_IL_u = [GM_l GM_u];
pm_IL_u = [-pm pm];

% Computing the Singular Values stability margins from Part 1.6, (I+L_u^(-1))
pm = 2*asin(sys_Lu_sigma1_min/2)*180/pi;
GM_l = 20*log10((1-sys_Lu_sigma1_min));
GM_u = 20*log10((1+sys_Lu_sigma1_min));
GM_IL_u_inv = [GM_l GM_u];
pm_IL_u_inv = [-pm pm];

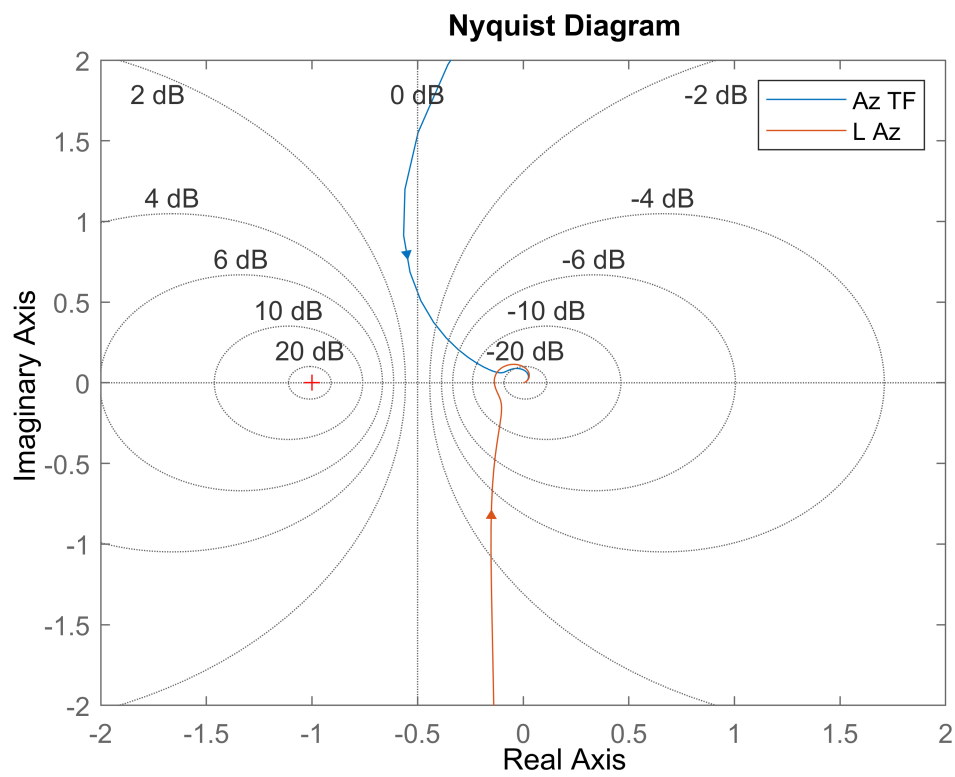
GM_sigma = [min(GM_IL_u_inv(1),GM_IL_u(1)),max(GM_IL_u_inv(2),GM_IL_u(2))];
PM_sigma = [min(pm_IL_u_inv(1),pm_IL_u(1)),max(pm_IL_u_inv(2),pm_IL_u(2))];

text = ['Classical Gain Margin: ',num2str(classical_GM_db(1)),' ',num2str(classical_GM_db(2)),' dB',
newline 'Classical Phase Margin: ',num2str(-classical_pm),' ',num2str(classical_pm),' degrees',
newline 'Singular Values Gain Margin: ',num2str(GM_sigma(1)),' ',num2str(GM_sigma(2)),' dB',
newline 'Singular Values Phase Margin: ',num2str(PM_sigma(1)),' ',num2str(PM_sigma(2)),' degrees'];
disp(text);
```

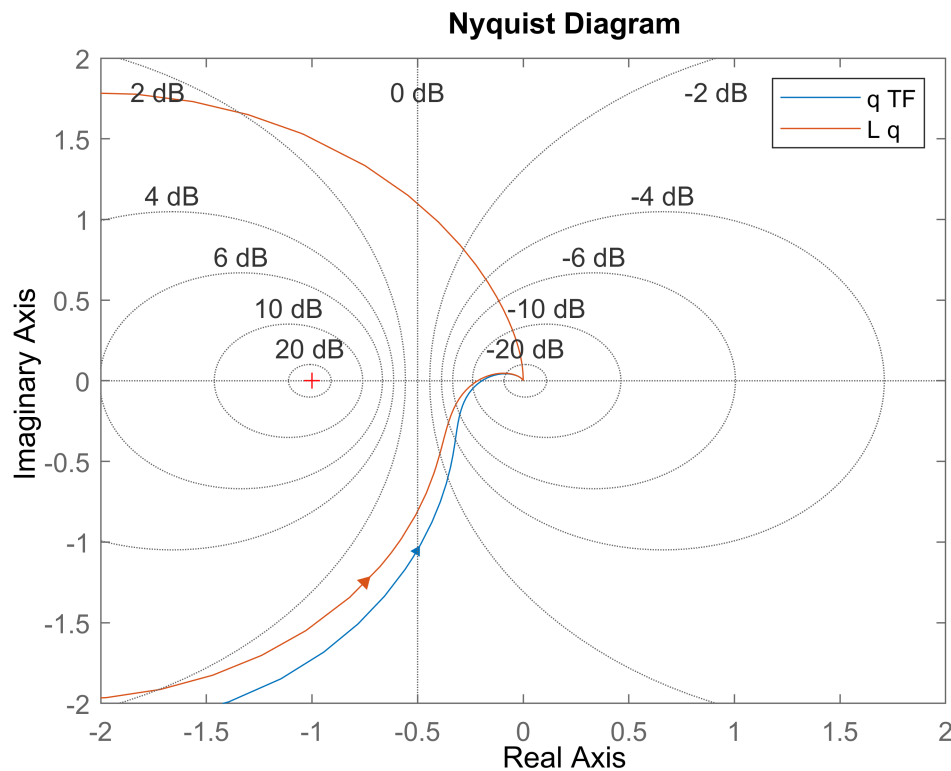
Classical Gain Margin: [-11.9938,10.6915] dB  
Classical Phase Margin: [-55.4905,55.4905] degrees  
Singular Values Gain Margin: [-11.4849,9.1594] dB  
Singular Values Phase Margin: [-43.0284,43.0284] degrees

## 2.9) Classic Stability Margins at the Plant Output $L_y$

```
% Plotting Nyquist Plots to visualize margins
n = nyquistplot(sys_Ly(1,1),Ly_Az_Open(1,1));
setoptions(n,'ShowfullContour','off'), grid on
legend('Az TF', 'L Az')
xlim([-2 2])
ylim([-2 2])
```



```
n = nyquistplot(sys_Ly(2,2),Ly_q_Open(2,2));
setoptions(n,'ShowfullContour','off'), grid on
legend('q TF', 'L q')
xlim([-2 2])
ylim([-2 2])
```



```
Ly_m_Az_tf = allmargin(sys_Ly(1,1));
Gm_classical_Az_tf = 20*log10(Ly_m_Az_tf.GainMargin);
Pm_classical_Az_tf = Ly_m_Az_tf.PhaseMargin;
Ly_m_q_tf = allmargin(sys_Ly(2,2));
Gm_classical_q_tf = 20*log10(Ly_m_q_tf.GainMargin);
Pm_classical_q_tf = Ly_m_q_tf.PhaseMargin;
text = ['Classical Gain Margin of Az TF: [-inf, inf] dB',...
newline 'Classical Phase Margin of Az TF: ',num2str(Pm_classical_Az_tf),'degrees',...
newline 'Classical Gain Margin of q TF: ',num2str(Gm_classical_q_tf(1)),', ',num2str(Gm_classical_q_tf(2)),...
newline 'Classical Phase Margin of q TF: ',num2str(Pm_classical_q_tf),'degrees'];
disp(text);
```

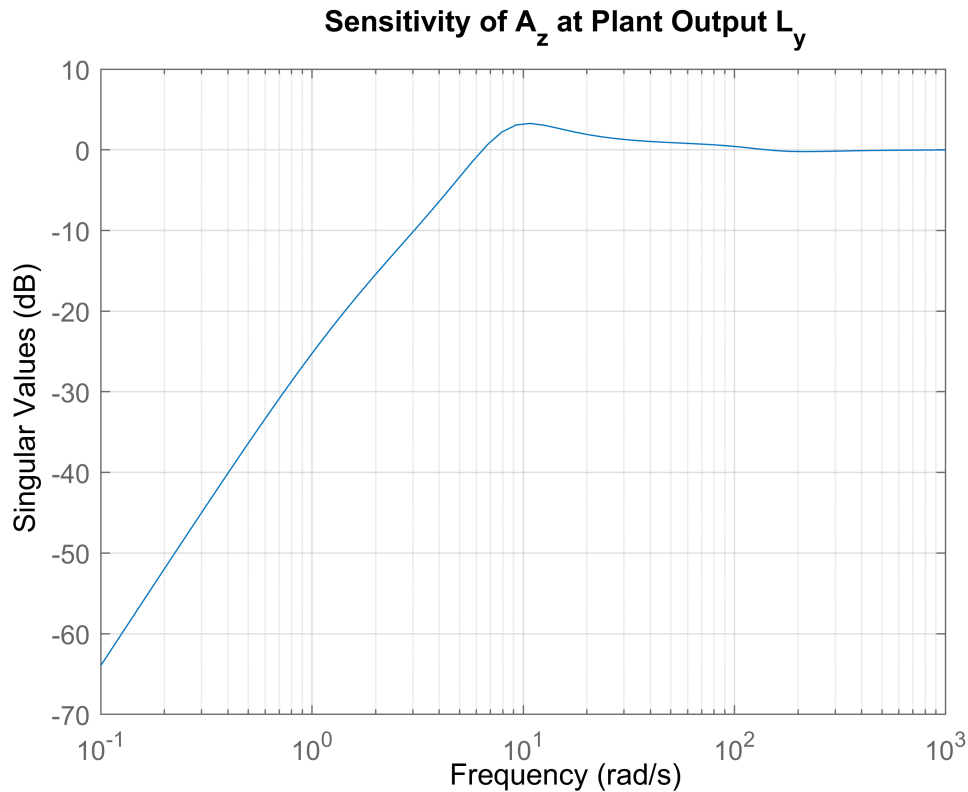
```
Classical Gain Margin of Az TF: [-inf, inf] dB
Classical Phase Margin of Az TF: -56.0122degrees
Classical Gain Margin of q TF: [-13.0436, 14.1755] dB
Classical Phase Margin of q TF: 63.9099degrees
```

The margins at the plant output are much wider than the margins found in part 1.8 especially the gain margins, this can be interpreted that large amounts of gain inserted at the plant output can be adjusted more without causing instability than inserting those gains at the plant input.

## 2.10) Plotting the Complementary Sensitivity T and Sensitivity S for the Acceleration Loop at Plant Output

```
Sensitivity = 1/(1+sys_Ly(1,1));
sigma(Sensitivity,{.1,1000})
```

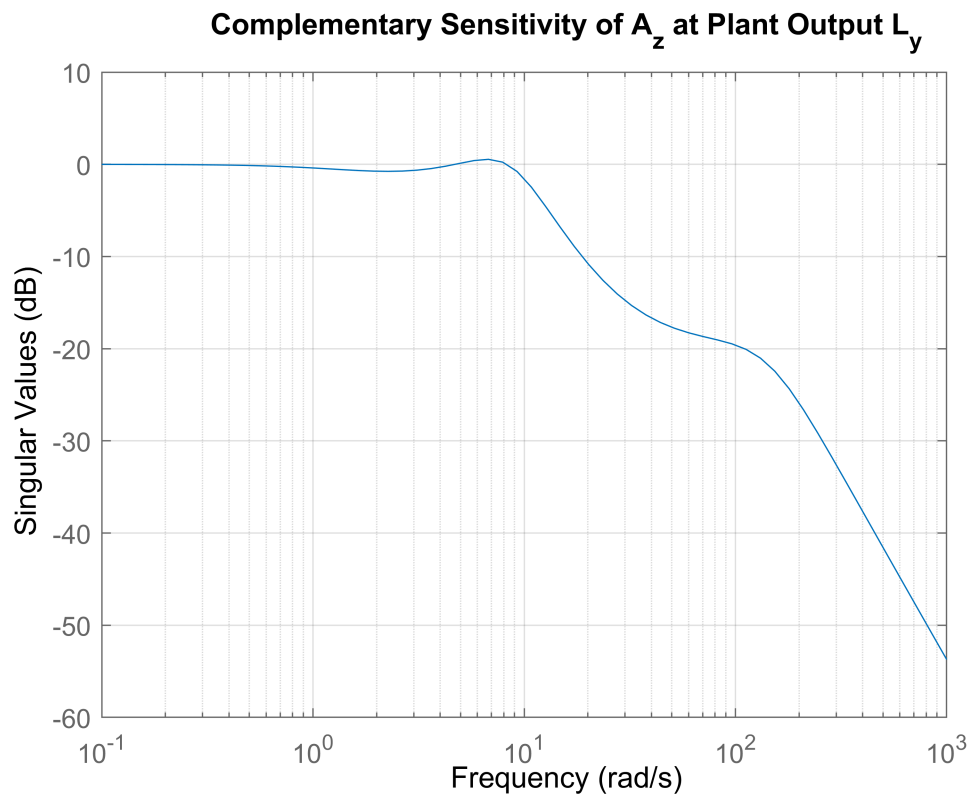
```
Sensitivity_max = max(sigma(Sensitivity,{.1,1000}));
grid on
title("Sensitivity of A_z at Plant Output L_y")
```



```
Sensitivity_max_db = 20*log10(Sensitivity_max);
text = ['Peak of The Sensitivity S is: ',num2str(Sensitivity_max),' or ',num2str(Sensitivity_max_db)];
disp(text);
```

Peak of The Sensitivity S is: 1.4575 or 3.2722 in db

```
Compl_Sensitivity = ((1+sys_Ly(1,1))^-1)*sys_Ly(1,1);
sigma(Compl_Sensitivity,{.1,1000})
Compl_Sensitivity_max = max(sigma(Compl_Sensitivity,{.1,1000}));
grid on
title("Complementary Sensitivity of A_z at Plant Output L_y")
```



```

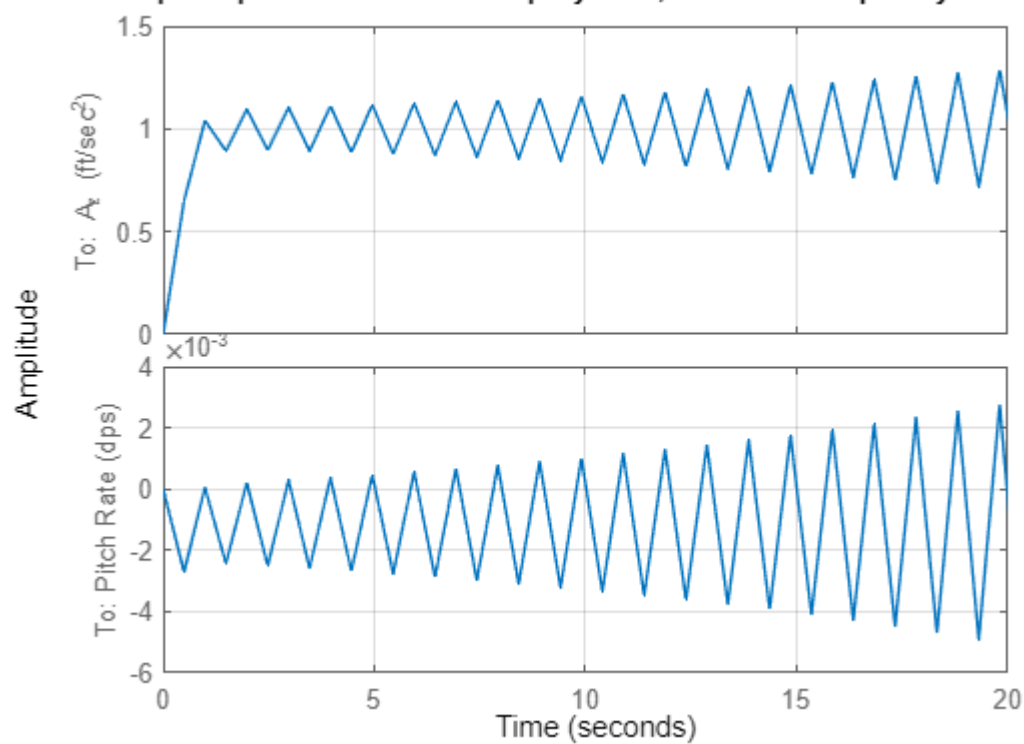
Compl_Sensitivity_max_db = 20*log10(Compl_Sensitivity_max);
text = ['Peak of The Complimentary Sensitivity T is: ',num2str(Compl_Sensitivity_max),' or ',num2str(Compl_Sensitivity_max_db),' dB'];
disp(text);

```

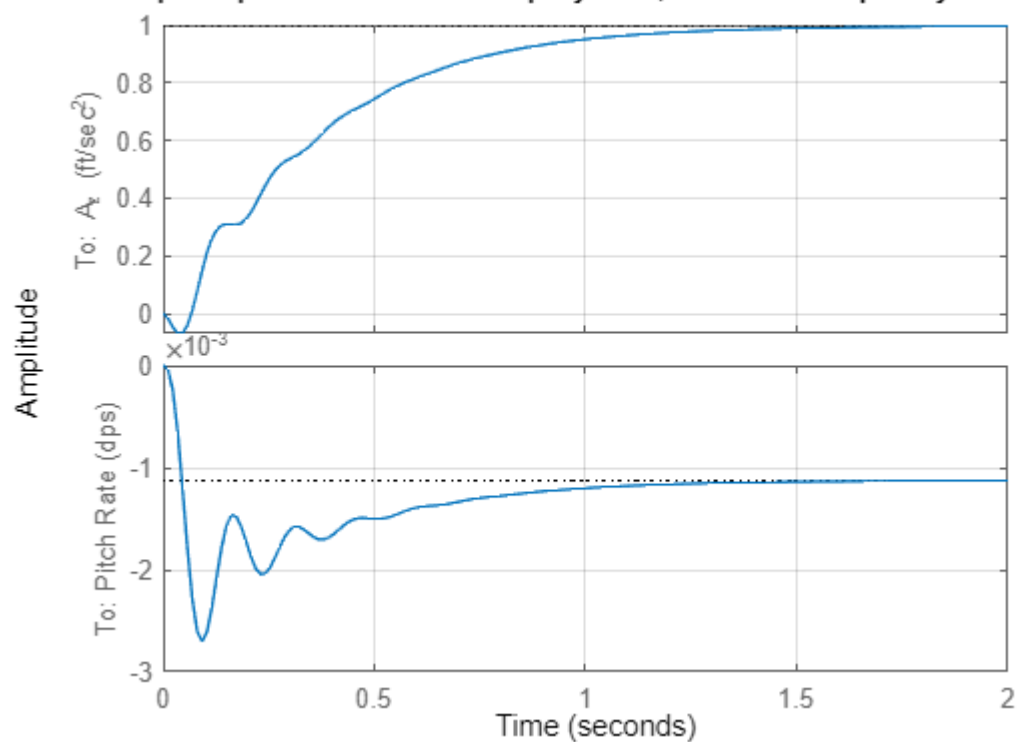
Peak of The Complimentary Sensitivity T is: 1.0647 or 0.54441 in db

**2.b) What happens to the stability and performance of the system if the actuator natural frequency is reduced to  $w_n = 60$  (rps)?**

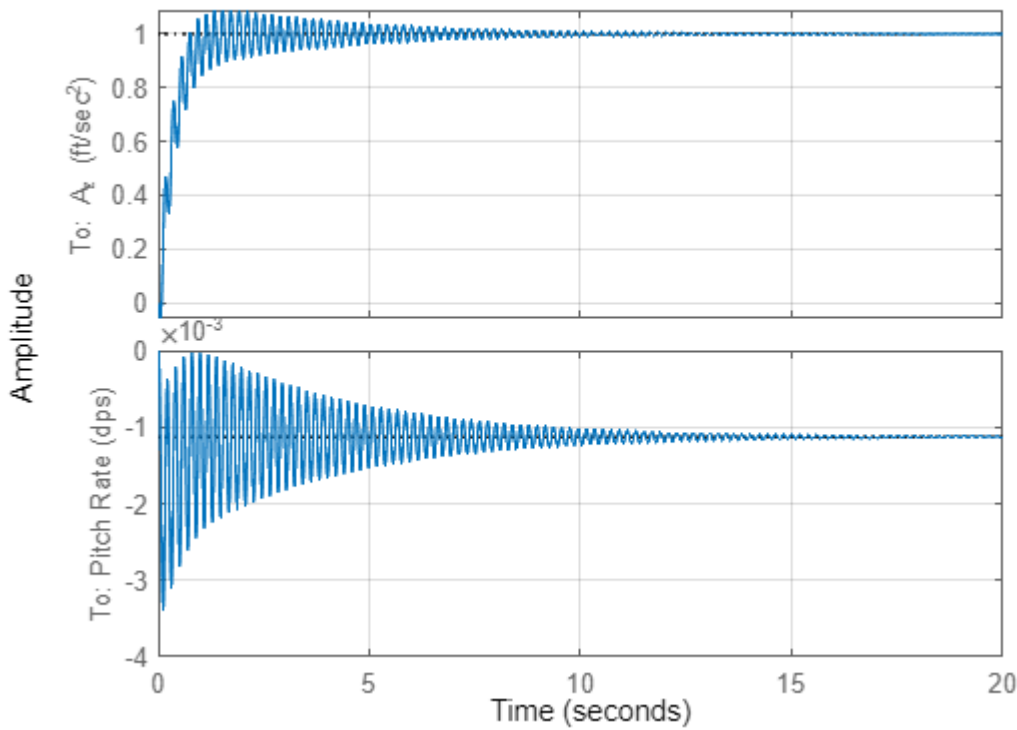
**Step Response of Closed Loop System, Actuator Frequency at 39 rps**



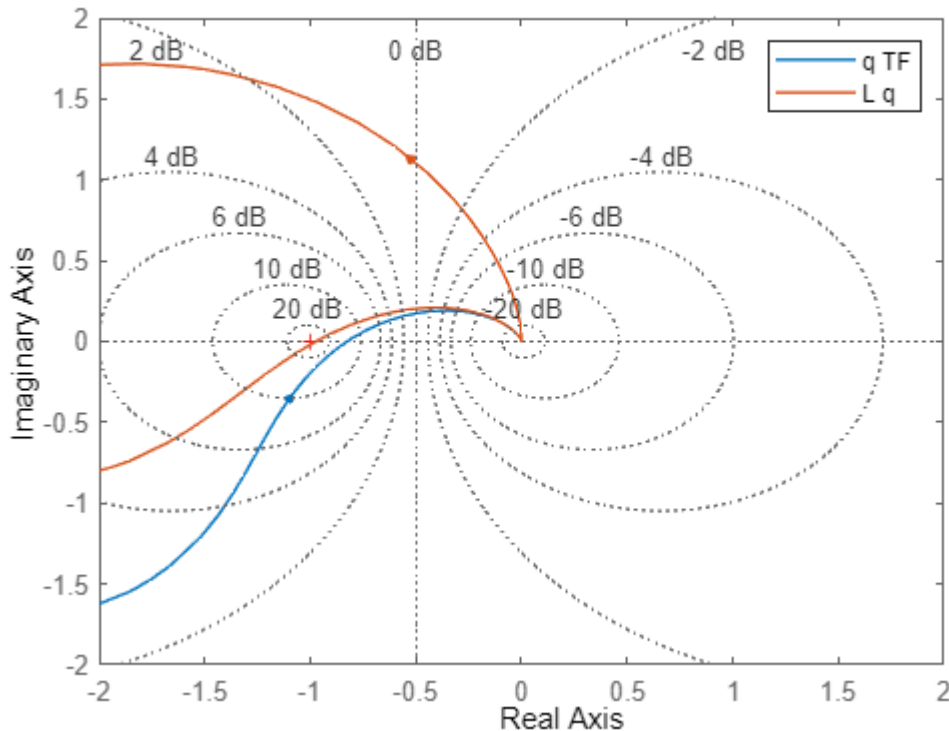
**Step Response of Closed Loop System, Actuator Frequency at 60 rps**



**Step Response of Closed Loop System, Actuator Frequency at 40 rps**



**Nyquist Plot of  $L_y$  at an actuator frequency of 40 rps**



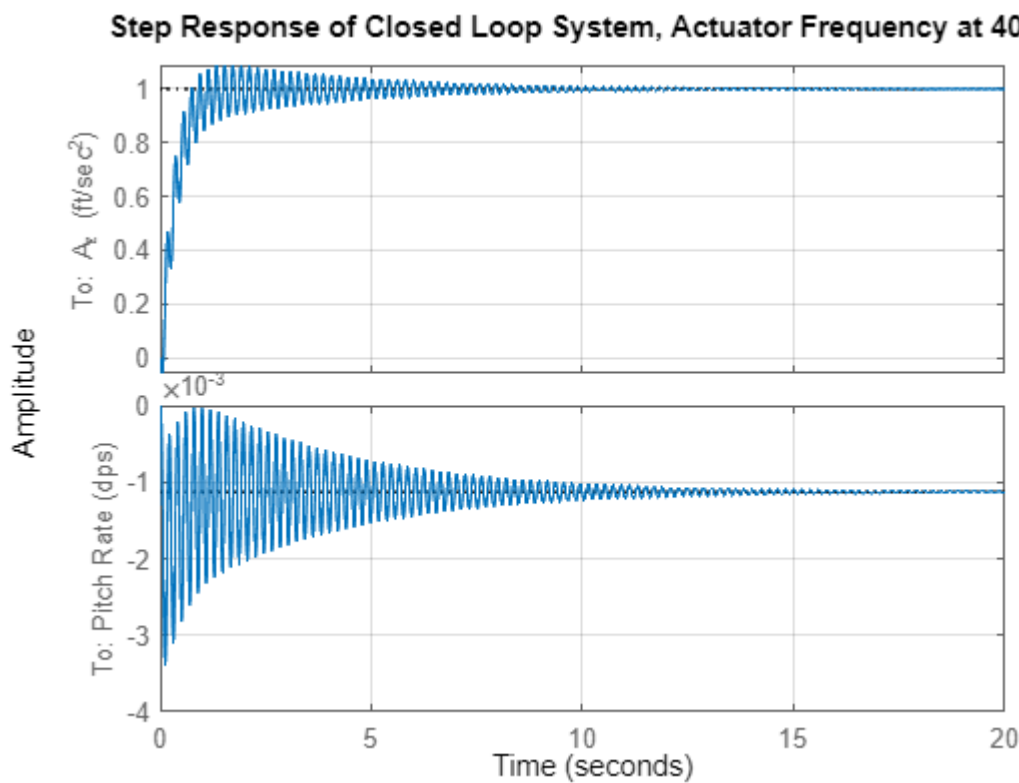
As can be seen from the plots above, the performance of the system is greatly reduced when the actuator frequency is reduced to 60 rps. As the actuator frequency decreases the stability margins take a hit, at 40 rps the nyquist plot above shows that the plot almost intersects the -1 showing how with the slightest change the system can make the system unstable. The change in the actuator frequency also affected the transient and

rise time of the system with it having a more negative affect on the pitch rate than  $A_z$ , The transient time increased significantly and so did the settling time, the rise time increased slightly for the  $A_z$  but got worse for  $q$ .

Table shows statistics of step response at 40 rps

| Fields | RiseTime | TransientTime | SettlingTime | SettlingMin | SettlingMax | Overshoot |
|--------|----------|---------------|--------------|-------------|-------------|-----------|
| 1      | 0.2466   | 3.8657        | 4.1539       | 0.5757      | 1.0891      | 8.90      |
| 2      | 0.0442   | 9.9415        | 12.5718      | -0.0034     | -2.6595e-05 | 200.90    |

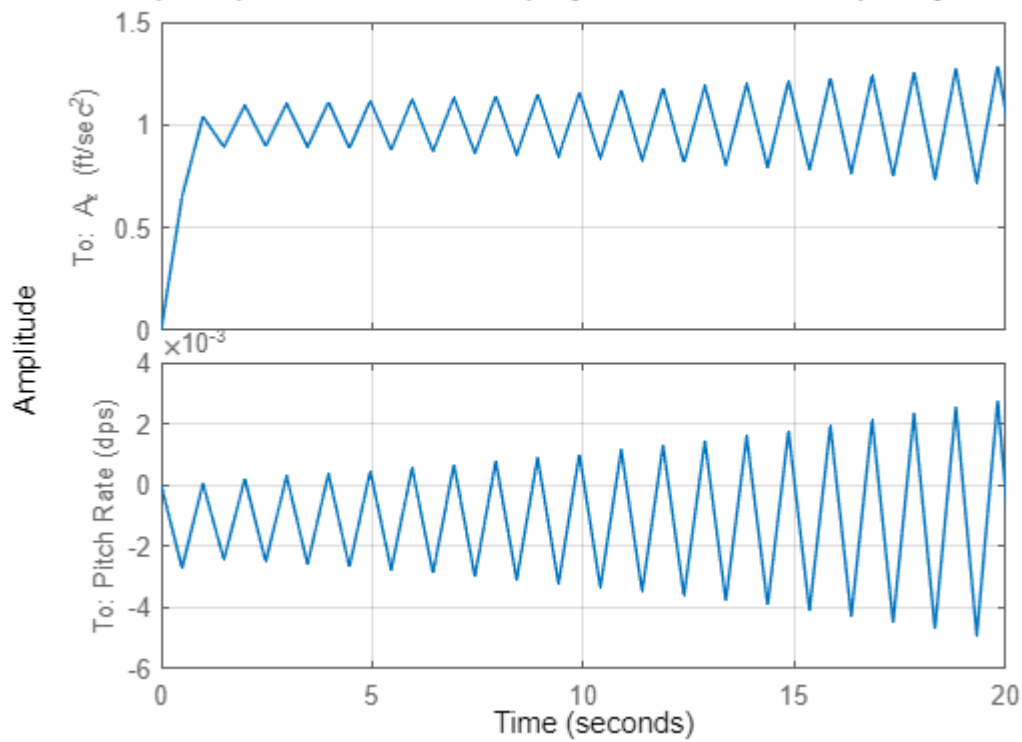
**2.b) What is the lowest actuator natural frequency (using these same gains) that produces a closed loop stable response?**



As can be seen from the above plot as well as from the previous question, the system is barely stable (marginally stable) at an actuator frequency of 40 rps and at 39 rps the system goes unstable as can be seen from the plot below, also the nyquist plot below shows how close the plot is to intersecting -1 and show that  $q$  is affected worse than  $A_z$ .



**Step Response of Closed Loop System, Actuator Frequency at 39 rps**



**Nyquist Plot of  $L_y$  at an actuator frequency of 40 rps**

