گزارش آزمایشگاه کنترل خطی (ربات با مفصل منعطف)

نیما حاجی حیدری (۹۹۲۳۰۱۷)، مهدی شاهینی (۹۹۲۳۰۴۰)

دانشگاه صنعتی امیرکبیر، دانشکده مهندسی برق

۱۴۰۱ - ۱۴۰۲ – ۲

فهرست مطالب

مقدمه 4

توصیف کلی سیستم 4

پاسخ سوالات بخش اول مبتنی بر کد و اعداد سری ۱ 6

طراحی کنترل کننده lead **برای کنترل رفتار مفصل نرم مبتنی بر کد ۱** 11

پیش طراحی 11

طراحی 12

پاسخ سوالات بخش اول مبتنی بر کد و اعداد سری ۲ 15

طراحی کنترل کننده lead **برای کنترل رفتار مفصل نرم مبتنی بر کد ۲** 18

پیش طراحی 18

طراحی 19

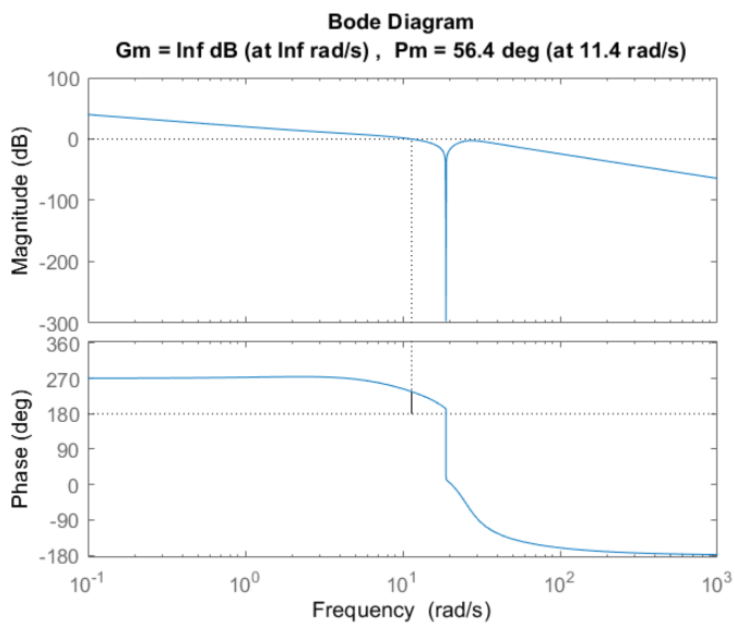
پاسخ سوالات: 22

طراحی کنترل کننده‌ی Lag 25

طراحی 25

Lag Controller

```
PM_lag = 75;
w_c_lag = 3.25;
beta_lag = 2.96;
z_lag = w_c_lag * 0.1;
p_lag = z_lag / beta_lag;
C_lag = (K_ctrl / beta_lag) * ((s+z_lag)/(s+p_lag));
T_ol_lag = series(C_lag, G_theta);
margin(T_ol)
```



27 ....

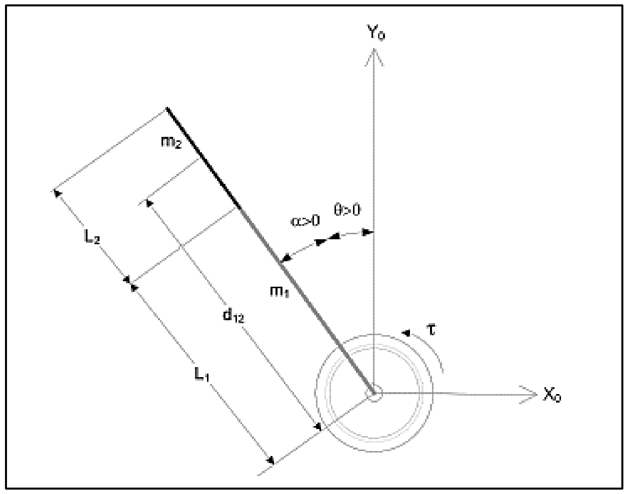
سوالات 28

طراحی کنترل‌کننده فیدبک در فضای حالت 28

طراحی 28

نتایج عملی 31

مقدمه



در این گزارش‌کار، اکثریت منبع تصاویر دیتاشیت این دستگاه است.

این گزارش‌کار، همراه با یک کد متلب پیش می‌رود که هر کد ادامه کد قبل است.

توصیف کلی سیستم

بازوی اصلی این ماژول بر روی چرخ دهنده‌ی تامین بار یک سروو موتور قرار می‌گیرد. زاویه‌ی سروو موتور یعنی θ، به طور قراردادی در جهت خلاف حرکت عقربه‌های ساعت، افزایش می‌یابد. پس نتیجه می‌گیریم هنگامی که ولتاژ کنترلی V_m مثبت باشد؛ سروو موتور (و به تبع آن بازوی مفصل) پادساعتگرد چرخش خواهند کرد.

همانطور که می‌توان دید؛ طول کل بازو می‌تواند با تغییر مکان قرارگیری روی چرخ دنده‌ی موتور، تغییر کند. بخش اصلی این بازو که مستقیماً به نقطه‌ی لولا وصل شده، طولی به اندازه‌ی L_1 و جرمی به اندازه‌ی m_1 دارد. بخش بالایی میله نیز متناظر با L_2 و m_2 می‌باشد. فاصله‌ی بین نقطه‌ی لولا و مرکز جرم بازوی بالایی (که در وسط آن قرار دارد)؛ با d_{12} مشخص می‌گردد. بدیهی است که کل بازو، لختی چرخشی J_1 داشته که البته با تغییر طول بازو، تغییر خواهد کرد. همچنین زاویه‌ی انحراف بازو از یک مرجع، با α نمایش داده شده و دارای جهت پادساعتگرد می‌باشد.

نکته حائز اهمیت این است که در آزمایشگاه اعداد بـر اسـاس، سـروو موتـور تنظـیم شـده طراحـی شـد امـا در گزارش‌کار علاوه بر سروو موتور آزمایشگاه، بر اساس دیتاشیت سروو موتور اصلی مفصل منعطف شـرکت کـوانزر یک کنترل‌کننده طراحی شد.)

برای آشنایی بیشتر با اعداد و مقیاس های مورد آزمایش ، یک نمونه از اطلاعات مفصل منعطف ارائه شده است:

$$B_{eq} = 0.004\ \frac{N \cdot m}{(rad/s)} \quad J_{eq} = 2.08 * 10^{-3}\ Kg \cdot m^2 \quad m_1 = 0.064\ Kg \quad L_1 = 0.298\ m$$

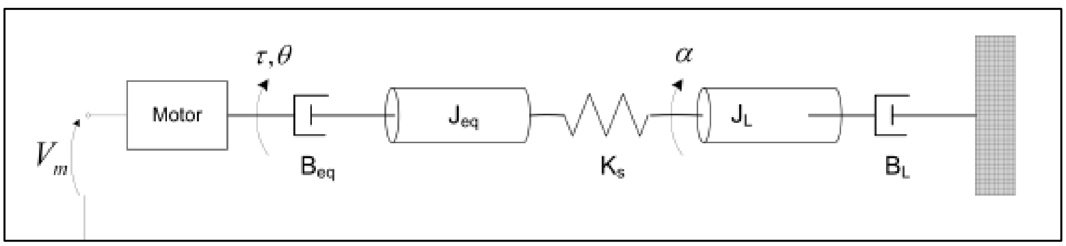
$$K_{stiff} = 1.25485 \quad J_{arm} = 0.0035\ Kg.m^2 \quad m_2 = 0.030\ Kg \quad L_2 = 0.156\ m$$

$$\eta_m = 0.69 \quad \eta_g = 0.9 \quad K_g = 70 \quad R_m = 2.6$$

$$K_t = K_m = 0.00767$$

نکته : ضریب B *در واقع ضریب اصطکاک وابسته به چرخش می‌باشد.*

میتوان سیستم مفصل را منعطف را در حالت کلی بصورت شماتیک زیر نمایش داد:



متغیر تحت کنترل ما، ولتاژ سروموتور یعنی همان V_m می‌باشد. این ولتاژ گشـتاوری مثـل τ در چـرخ دنـده‌ی سروو موتور تولید می‌کند تا بتواند بازو را بچرخاند. ضریب اصطکاک ویسکوز موتور به صورت معادل با B_{eq} مدل می‌گردد که به تبع، گشتاور مخالفی با گشتاور موثر موتور تولید می‌نماید. ریب اصطکاک ویسکوز مربوط به بـازو نیز با B_l مشخص شده و در نهایت بازو می‌تواند به صورت یک فنر با ضریب سختی K_s مدل شود.

مبتنی بر کد ۱

پاسخ سوالات بخش اول مبتنی بر کد و اعداد سری ۱

+ معادله نهایی توصیف‌کننده K_{stiff} را بدست آورید.

- اگر از ممان در $\alpha = 0$ مشتق بگیریم؛ خواهیم داشت:

$$M = M_x + M_y = R\left[(F_{2x} - F_{1x})\cos(\alpha) - \left(F_{2y} + F_{1y}\right)\sin(\alpha)\right]$$

$$\begin{aligned} K_{stiff} &= \left(\frac{\partial M}{\partial \alpha}\right)\Big|_{\alpha=0} \\ &= \left(R\left[(F_{1x} - F_{2x})\sin(\alpha) - \left(F_{2y} + F_{1y}\right)\cos\left(\alpha\left(F_{2y} + F_{1y}\right)\right)\right]\right)\Big|_{\alpha=0} \\ &= -R\left(F_{2y} + F_{1y}\right) = -RF_y \end{aligned}$$

$$\boxed{K_{stiff} = -RF_y}$$

+ با فرض در دسترس بودن ممان چرخشی بدنه و همچنین ممان چرخشی بازو (J_{hub} و J_{arm}) انرژی پتانسیل و جنبشی کلی سیستم را محاسبه کنید.

- تعریف و فرمول انرژی جنبشی هر بخش از سیستم را می‌نویسیم:

$$K_{Total} = K_{hub} + K_{arm} = \frac{1}{2}J_{hub}\dot{\theta}^2 + \frac{1}{2}J_{arm}\left(\dot{\theta} + \dot{\alpha}\right)^2$$

$$P = \frac{1}{2}K_{stiff}\alpha^2$$

+ ترم $B_{eq}\dot{\theta}$ بیانگر چه پارامتری در سیستم می‌باشد؟

- نشانگر گشتاور منفی ناشی از اصطکاک جرم و میله است.

+ ماتریس‌های فضای حالت را بدست آورید.

$$A=\begin{bmatrix} 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ 0 & J_{hub}^{-1}K_{stiff} & -J_{hub}^{-1}\left(\frac{r\eta_m K_t K_m}{R_m}+B_{eq}\right) & 0\\ 0 & -(J_{arm}^{-1}+J_{hub}^{-1})K_{stiff} & J_{hub}^{-1}\left(\frac{r\eta_m K_t K_m}{R_m}+B_{eq}\right) & 0\end{bmatrix}$$

$$B=J_{hub}^{-1}\left(\frac{r\eta_m K_t}{R_m}\right)\begin{bmatrix}0\\0\\1\\-1\end{bmatrix},\qquad C=\begin{bmatrix}1&0&0&0\\0&1&0&0\end{bmatrix},\qquad D=\begin{bmatrix}0\\0\end{bmatrix}$$

+ قطب‌های سیستم را از ماتریس A بدست آورید.

$$s_p=|\lambda I-A| == Eigenvalues\ of\ "A"$$

```
close
clear
clc
s = tf('s');
```

Initializing

```
eta_g = 0.9;
K_g = 70;
r = eta_g*K_g;
eta_m = 0.69;
K_t = 0.00767;
K_m = 0.00767;
K_stiff = 1.2485;
R_m = 2.6;
J_arm = 0.0035;
J_hub = 0.0026;
J_eq = 0.00208;
B_eq = 0.004;
m_1 = 0.064;
m_2 = 0.030;
L_1 = 0.298;
L_2 = 0.156;
```

A

```
A11 = zeros(2);
A12 = eye(2);
A21 = [0                inv(J_hub)*K_stiff;
       0 -(inv(J_hub) + inv(J_arm))*K_stiff;];
A22 = [-inv(J_hub)*(((r*eta_m*K_t*K_m)/R_m) + B_eq) 0;
       +inv(J_hub)*(((r*eta_m*K_t*K_m)/R_m) + B_eq) 0];
A = [A11 A12;
     A21 A22];
display(A)
```

```
A = 4×4
         0         0    1.0000         0
         0         0         0    1.0000
         0  480.1923   -1.9168         0
         0 -836.9066    1.9168         0
```

Poles

```
sys_p = eig(A);
display(sys_p)
```

```
sys_p = 4×1 complex
   0.0000 + 0.0000i
  -0.5495 +28.9086i
  -0.5495 -28.9086i
  -0.8179 + 0.0000i
```

+ توابع تبدیل را بدست آورید.

$$Y = C(SI - A)^{-1}BU + D$$

B

```
B = ((r*eta_m*K_t*K_m)/(J_hub*R_m))*[0;
                                      0;
                                      1;
                                     -1];
display(B)
```

```
B = 4×1
        0
        0
   0.3783
  -0.3783
```

C

```
C = [eye(2) zeros(2)];
display(C)
```

```
C = 2×4
     1     0     0     0
     0     1     0     0
```

D and U

```
D = [0;
     0];
display(D)
```

```
D = 2×1
     0
     0
```

```
U = 1;
display(U)
```

```
U = 1
```

Transfer Function

```
[Num , Den] = ss2tf(A, B, C, D);
display(Num), display(Den)
```

```
Num = 2×5
         0         0    0.3783         0  134.9442
         0         0   -0.3783   -0.0000   -0.0000

Den = 1×5
    1.0000    1.9168  836.9066  683.7354         0
```

```
G_theta = tf(Num(1,:), Den);
G_theta_pole = pole(G_theta);
G_alfa = tf(Num(2,:), Den);
G_alfa_pole = pole(G_alfa);
display(G_theta), display(G_theta_pole)
```

```
G_theta =

          0.3783 s^2 + 134.9
  -------------------------------------
  s^4 + 1.917 s^3 + 836.9 s^2 + 683.7 s

Continuous-time transfer function.
 G_theta_pole = 4×1 complex
    0.0000 + 0.0000i
   -0.5495 +28.9086i
   -0.5495 -28.9086i
   -0.8179 + 0.0000i
```

```
51    display(G_alfa), display(G_alfa_pole)

G_alfa =

  -0.3783 s^2 - 1.68e-16 s - 1.545e-32
  ------------------------------------
  s^4 + 1.917 s^3 + 836.9 s^2 + 683.7 s

Continuous-time transfer function.
 G_alfa_pole = 4×1 complex
      0.0000 + 0.0000i
     -0.5495 +28.9086i
     -0.5495 -28.9086i
     -0.8179 + 0.0000i
```

+ آیا در این سیستم قطب‌ها و مقادیر ویژه برابرند؟ توضیح دهید.

- بله. همانطور که نشان داده شده است قطب‌ها با مقادیر ویژه برابرند زیرا در صورتی که حذف صفر و قطب رخ ندهد شاهد این خواهیم بود که مقادیر ویژه ماتریس حالت، قطب‌های سیستم خواهد بود.

طراحی کنترل کننده lead برای کنترل رفتار مفصل نرم مبتنی بر کد ۱

پیش طراحی

ابتدا باید از کنترل‌پذیر بودن سیستم، اطمینان حاصل کرد. این به آن معنا است که باید ماتریس کنترل‌پذیری را تشکیل داد. حال که سیستم ما از مرتبه ۲ بوده و چهار متغیر حالت دارد؛ خواهیم داشت:

$$L_c = [B \quad AB \quad A^2B \quad A^3B]$$

البته این ماتریس با دستور خاص خود در محیط متلب، آرایش داده می‌شود. می‌دانیم که باید رنک ماتریس کنترل‌پذیری در این سیستم، مساوی با ۴ باشد. با چک کردن در توابع مربوط، درمی‌یابیم که سیستم کنترل‌پذیر است.

Controlability Check

```
C_c = ctrb(A, B);
C_c_rank = rank(C_c);
if C_c_rank == length(A)
    disp("It is controlable!"), disp(C_c_rank)
else
    disp("It is not controlable!"), display(C_c_rank), display(length(A))
end
```

```
It is controlable!
     4
```

طبق صورت سوال ، باید کنترل‌کننده‌ای طراحی گردد که دارای حد فاز مطلوب ۷۰ درجه بوده و ثابت خطای شیب آن ۱۰ باشد. همچنین پهنای باند ۵ رادیان بر ثانیه نیز جزء مشخصات مطلوب می‌باشد.

طراحی

میدانیم که فرم کلی یک کنترل‌کننده پیش‌فاز به صورت روبرو خواهد بود:

$$G_c = K_c \alpha \frac{s+z}{s+p}$$

که در آن ، α همان نسبت قطب به صفر می‌باشد. به علت آن‌که می‌خواهیم قطب روی سیستم ما تاثیر نگذارد؛ آن را بسیار دورتر از صفر در نظر می‌گیریم. برای ثابت خطای شیب خواهیم داشت:

$$K_v = \lim_{s \to 0} s \cdot G_c(s). G(s)$$

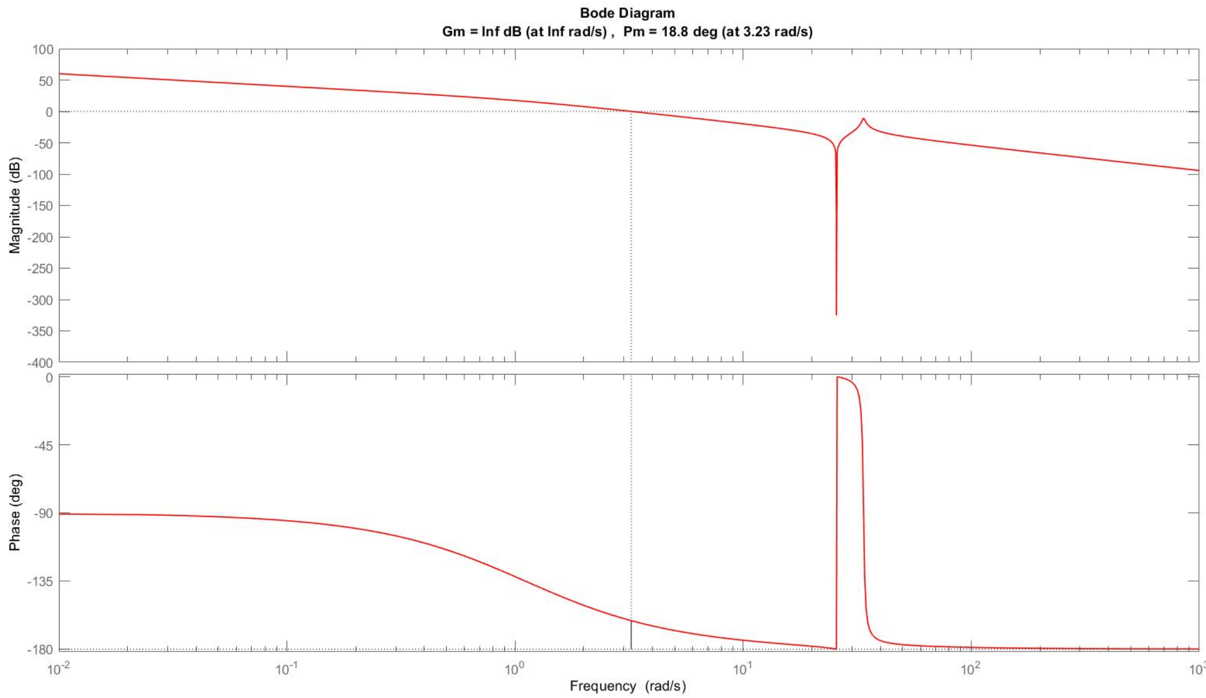
K as Gain

```
K_v = Num(9) / Den(4);
K_ctrl = 10 / K_v;
display(K_v), display(K_ctrl)
```

```
K_v = 0.1974
K_ctrl = 50.6680
```

حال که بهره تنظیم شد؛ دیاگرام Bode سیستم جبران نشده که تنها بهره آن تعیین شده است را رسم می‌کنیم و حد فاز آن را می‌یابیم. این کار به این صورت انجام می‌شود که فرکانسی که در آن اندازه صفر شده را یافته و می‌بینیم که فاز سیستم در آن فرکانس چقدر با ۱۸۰- درجه فاصله دارد . طبق دیاگرام‌هایی که در زیر آمده است؛ خواهیم داشت :

$$\phi_m = \phi_d - \phi_{sys} + 5° = 70° - 53° + 9.7° \rightarrow \boxed{\phi_m = 27°}$$



حال باید ضریب α را بیابیم. می‌دانیم که رابطه‌ی بین α و میزان افزایش فاز سیستم به صورت زیر است:

$$\sin\phi_m = \frac{\alpha-1}{\alpha+1} \rightarrow \sin 60° = \frac{\sqrt{3}}{2} \rightarrow \boxed{\alpha = 13.93}$$

سپس باید فرکانس فعالیت سیستم را پیدا کنیم. در این فرکانس باید دامنه‌ی سیستم جبران نشده که دیاگرام Bode آن را رسم کردیم؛ برابر با $10\log\alpha$ باشد. از آنجایی که $\alpha = 13.93$ است پس خواهیم داشت:

$$\boxed{\omega_m = 1.55\ rad/s}$$

در مرحله بعد از ترکیب دو معادله و دو مجهول می‌توان صفر و قطب سیستم را یافت. داریم:

$$\begin{cases} p = \sqrt{\omega_m^2 * \alpha} \\ z = \dfrac{p}{\alpha} \end{cases} \rightarrow \boxed{p = 5.785, z = 0.4153}$$

حال به تابع تبدیل کنترل‌کننده بازگشته و مقادیر یافت شده را در آن قرار می‌دهیم. که با خروجی ما تناسب ندارد(فیز مارجین ۳۵ درجه!). حال به سراغ ماتریس‌های معرفی شده توسط مربی آزمایشگاه می‌رویم.

باتوجه به خروجی‌های بدست آمده از اعداد کد ۱،

به سراغ کد ۲ می‌رویم و محاسبات را تکرار می‌کنیم.

پاسخ سوالات بخش اول مبتنی بر کد و اعداد سری ۲

+ معادله نهایی توصیف‌کننده K_{stiff} را بدست آورید.

- اگر از ممان در $\alpha = 0$ مشتق بگیریم؛ خواهیم داشت:

$$M = M_x + M_y = R\left[(F_{2x} - F_{1x})\cos(\alpha) - \left(F_{2y} + F_{1y}\right)\sin(\alpha)\right]$$

$$\begin{aligned} K_{stiff} &= \left(\frac{\partial M}{\partial \alpha}\right)\bigg|_{\alpha=0} \\ &= \left(R\left[(F_{1x} - F_{2x})\sin(\alpha) - \left(F_{2y} + F_{1y}\right)\cos\left(\alpha\left(F_{2y} + F_{1y}\right)\right)\right]\right)\bigg|_{\alpha=0} \\ &= -R\left(F_{2y} + F_{1y}\right) = -RF_y \end{aligned}$$

$$\boxed{K_{stiff} = -RF_y}$$

+ با فرض در دسترس بودن ممان چرخشی بدنه و همچنین ممان چرخشی بازو (J_{hub} و J_{arm}) انرژی پتانسیل و جنبشی کلی سیستم را محاسبه کنید.

- تعریف و فرمول انرژی جنبشی هر بخش از سیستم را می‌نویسیم:

$$K_{Total} = K_{hub} + K_{arm} = \frac{1}{2}J_{hub}\dot{\theta}^2 + \frac{1}{2}J_{arm}\left(\dot{\theta} + \dot{\alpha}\right)^2$$

$$P = \frac{1}{2}K_{stiff}\alpha^2$$

+ ترم $B_{eq}\dot{\theta}$ بیانگر چه پارامتری در سیستم می‌باشد؟

- نشانگر گشتاور منفی ناشی از اصطکاک جرم و میله است.

+ ماتریس‌های فضای حالت را بدست آورید.

$$A=\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & J_{hub}^{-1}K_{stiff} & -J_{hub}^{-1}\left(\dfrac{r\eta_m K_t K_m}{R_m}+B_{eq}\right) & 0 \\ 0 & -(J_{arm}^{-1}+J_{hub}^{-1})K_{stiff} & J_{hub}^{-1}\left(\dfrac{r\eta_m K_t K_m}{R_m}+B_{eq}\right) & 0 \end{bmatrix}$$

$$B=J_{hub}^{-1}\left(\frac{r\eta_m K_t}{R_m}\right)\begin{bmatrix}0\\0\\1\\-1\end{bmatrix},\qquad C=\begin{bmatrix}1&0&0&0\\0&1&0&0\end{bmatrix},\qquad D=\begin{bmatrix}0\\0\end{bmatrix}$$

+ قطب‌های سیستم را از ماتریس A بدست آورید.

$$s_p=|\lambda I-A| == Eigenvalues\ of\ "A"$$

اطلاعات را وارد می‌کنیم و ماتریس A را مستقیما بدست می‌آوریم.

```
close
clear
clc
s = tf('s');
```

A

```
A = [0      0       1 0;
        0      0       0 1;
        0 483.13 -28.262 0;
        0 -837.21 28.262 0];
display(A)
```

```
A = 4×4
            0         0    1.0000         0
            0         0         0    1.0000
            0  483.1300  -28.2620         0
            0 -837.2100   28.2620         0
```

Poles

```
sys_p = eig(A);
display(sys_p)
```

```
sys_p = 4×1 complex
     0.0000 + 0.0000i
    -6.3106 +24.4945i
    -6.3106 -24.4945i
   -15.6407 + 0.0000i
```

+ توابع تبدیل را بدست آورید.

$$Y = C(SI - A)^{-1}BU + D$$

B

```
B = [0;
     0;
     49.708;
     -49.708];
display(B)
```

```
B = 4×1
         0
         0
   49.7080
  -49.7080
```

C

```
C = [eye(2) zeros(2)];
display(C)
```

```
C = 2×4
     1     0     0     0
     0     1     0     0
```

D and U

```
D = [0;
     0];
display(D)
```

```
D = 2×1
     0
     0
```

```
U = 1;
display(U)
```

```
U = 1
```

Transfer Function

```
[Num , Den] = ss2tf(A, B, C, D);
display(Num), display(Den)
```

```
Num = 2×5
10^4 ×
          0          0     0.0050     0.0000     1.7601
          0          0    -0.0050     0.0000    -0.0000

Den = 1×5
10^4 ×
     0.0001     0.0028     0.0837     1.0007          0
```

```
G_theta = tf(Num(1,:), Den);
G_theta_pole = pole(G_theta);
G_alfa = tf(Num(2,:), Den);
G_alfa_pole = pole(G_alfa);
display(G_theta), display(G_theta_pole)
```

```
G_theta =

     49.71 s^2 + 1.004e-26 s + 1.76e04
  ----------------------------------------
  s^4 + 28.26 s^3 + 837.2 s^2 + 1.001e04 s

Continuous-time transfer function.
G_theta_pole = 4×1 complex
    0.0000 + 0.0000i
   -6.3106 +24.4945i
   -6.3106 -24.4945i
  -15.6407 + 0.0000i
```

```
display(G_alfa), display(G_alfa_pole)
```

```
G_alfa =

    -49.71 s^2 + 3.532e-13 s - 5.198e-28
  ----------------------------------------
  s^4 + 28.26 s^3 + 837.2 s^2 + 1.001e04 s

Continuous-time transfer function.
G_alfa_pole = 4×1 complex
    0.0000 + 0.0000i
   -6.3106 +24.4945i
   -6.3106 -24.4945i
  -15.6407 + 0.0000i
```

+ آیا در این سیستم قطب‌ها و مقادیر ویژه برابرند؟ توضیح دهید.

- بله. همانطور که نشان داده شده است قطب‌ها با مقادیر ویژه برابرند زیرا در صورتی که حذف صفر و قطب رخ ندهد شاهد این خواهیم بود که مقادیر ویژه ماتریس حالت، قطب‌های سیستم خواهند بود.

طراحی کنترل کننده lead برای کنترل رفتار مفصل نرم مبتنی بر کد ۲

پیش طراحی

ابتدا باید از کنترل‌پذیر بودن سیستم، اطمینان حاصل کرد. این به آن معنا است که باید ماتریس کنترل‌پذیری را تشکیل داد. حال که سیستم ما از مرتبه ۲ بوده و چهار متغیر حالت دارد؛ خواهیم داشت:

$$L_c = [B \quad AB \quad A^2B \quad A^3B]$$

البته این ماتریس با دستور خاص خود در محیط متلب، آرایش داده می‌شود. می‌دانیم که باید رنک ماتریس کنترل‌پذیری در این سیستم، مساوی با ۴ باشد. با چک کردن در توابع مربوط، درمی‌یابیم که سیستم کنترل‌پذیر است.

Controlability Check

```
C_c = ctrb(A, B);
C_c_rank = rank(C_c);
if C_c_rank == length(A)
    disp("It is controlable!"), disp(C_c_rank)
else
    disp("It is not controlable!"), display(C_c_rank), display(length(A))
end
```

```
It is controlable!
     4
```

طبق صورت سوال ، باید کنترل‌کننده‌ای طراحی گردد که دارای حد فاز مطلوب ۷۰ درجه بوده و ثابت خطای شیب آن ۱۰ باشد. همچنین پهنای باند ۵ رادیان بر ثانیه نیز جزء مشخصات مطلوب می‌باشد.

طراحی

میدانیم که فرم کلی یک کنترل‌کننده پیش‌فاز به صورت روبرو خواهد بود:

$$G_c = K_c \alpha \frac{s+z}{s+p}$$

که در آن ، α همان نسبت قطب به صفر می‌باشد. به علت آن‌که می‌خواهیم قطب روی سیستم ما تاثیر نگذارد؛ آن را بسیار دورتر از صفر در نظر می‌گیریم. برای ثابت خطای شیب خواهیم داشت:

$$K_v = \lim_{s \to 0} s \cdot G_c(s).G(s)$$

K as Gain

```
K_v = Num(9) / Den(4);
K_ctrl = 10 / K_v;
display(K_v), display(K_ctrl)
```

```
K_v = 1.7588
K_ctrl = 5.6856
```

حال که بهره تنظیم شد؛ دیاگرام Bode سیستم جبران نشده که تنها بهره آن تعیین شده است را رسم می‌کنیم و حد فاز آن را می‌یابیم. این کار به این صورت انجام می‌شود که فرکانسی که در آن اندازه صفر شده را یافته و می‌بینیم که فاز سیستم در آن فرکانس چقدر با ۱۸۰- درجه فاصله دارد. خواهیم داشت:

```
[Gm, Pm] = margin(K_ctrl*G_theta);
```

```
Gm = Inf
Pm = 53.0221
```

$$\phi_m = \phi_d - \phi_{sys} + 5° = 70° - 53° + 10° \rightarrow \boxed{\phi_m = 22°}$$

$$\sin \phi_m = \frac{\alpha - 1}{\alpha + 1} \rightarrow \sin 22° \rightarrow \boxed{\alpha = 2.61}$$

سپس باید فرکانس فعالیت سیستم را پیدا کنیم. در این فرکانس باید دامنه‌ی سیستم جبران نشده که دیاگرام Bode آن را رسم کردیم؛ برابر با $10\log\alpha$ باشد. از آنجایی که $\alpha = 2.61$ است پس خواهیم داشت:

$$\boxed{\omega_m = 5.56\ rad/s}$$

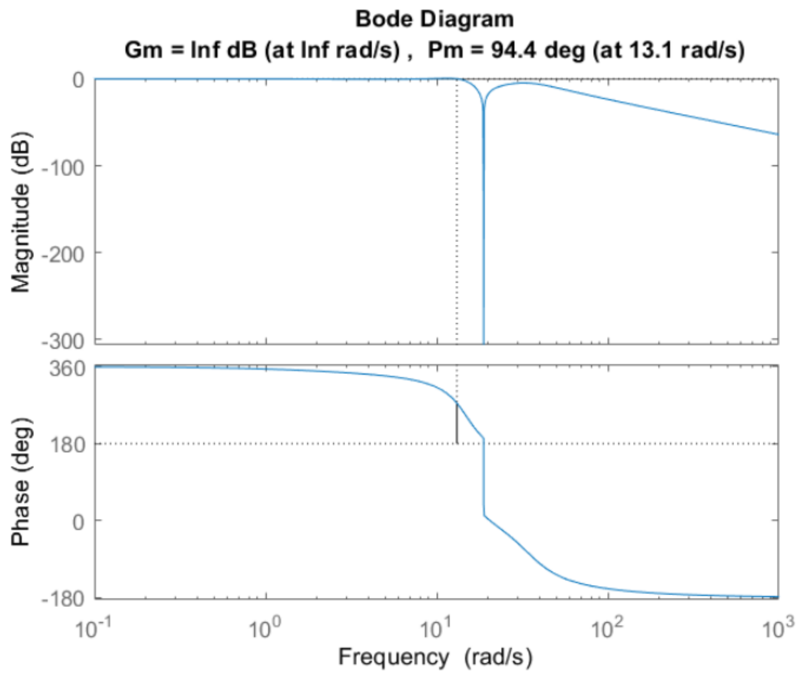
در مرحله بعد از ترکیب دو معادله و دو مجهول می‌توان صفر و قطب سیستم را یافت. داریم:

$$\begin{cases} p = \sqrt{\omega_m^2 * \alpha} \\ z = \dfrac{p}{\alpha} \end{cases} \rightarrow \boxed{p = 9, z = 3.44}$$

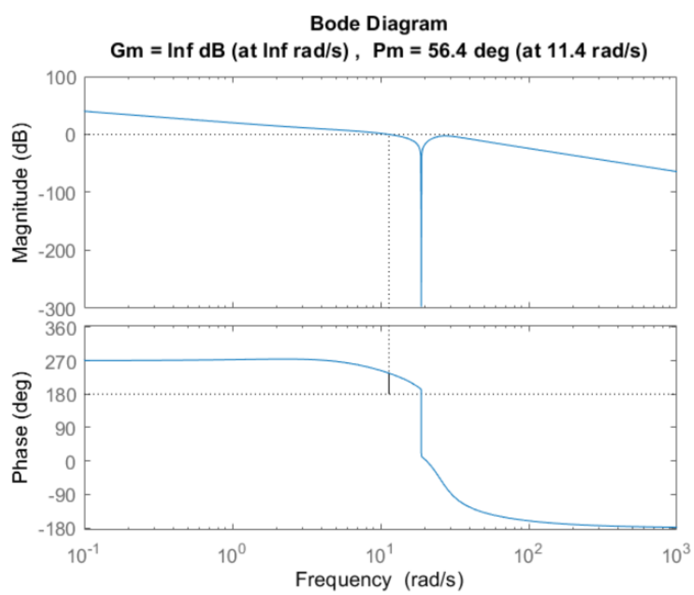
حال به تابع تبدیل کنترل‌کننده بازگشته و مقادیر یافت شده را در آن قرار می‌دهیم.

```
w_m = 5.9
w_m = 5.9000
p = sqrt(alfa * w_m^2)
p = 8.7471
z = p / alfa
z = 3.9796
```

```
T_ol = K_ctrl * alfa * ((s+z)/(s+p)) * G_theta;
T_cl = feedback(T_ol, 1);
margin(T_cl)
```




```
margin(T_ol)
```



پاسخ سوالات:

+ دلیل افزودن ۵ درجه؟

- زیرا زمانی که به فاز سیستم افزوده می‌شود ω_c کمی جابه‌جا می‌شود. این جابه‌جا شدن سبب می‌شود که افزایش فاز کمتر از حد مطلوب باشد پس آن را ۵ درجه بیشتر می‌افزاییم تا در صورت جابه‌جا شدن ω_c به مقدار مطلوب برسیم.

+ چرا گین کنترلی در محاسبه افزایش اندازه تاثیری ندارد؟

- زیرا نمودار bode رسم شده؛ خود تابع تبدیل در گین کنترلی بود و دیگر نیازی به محاسبه در کنترلر ندارد.

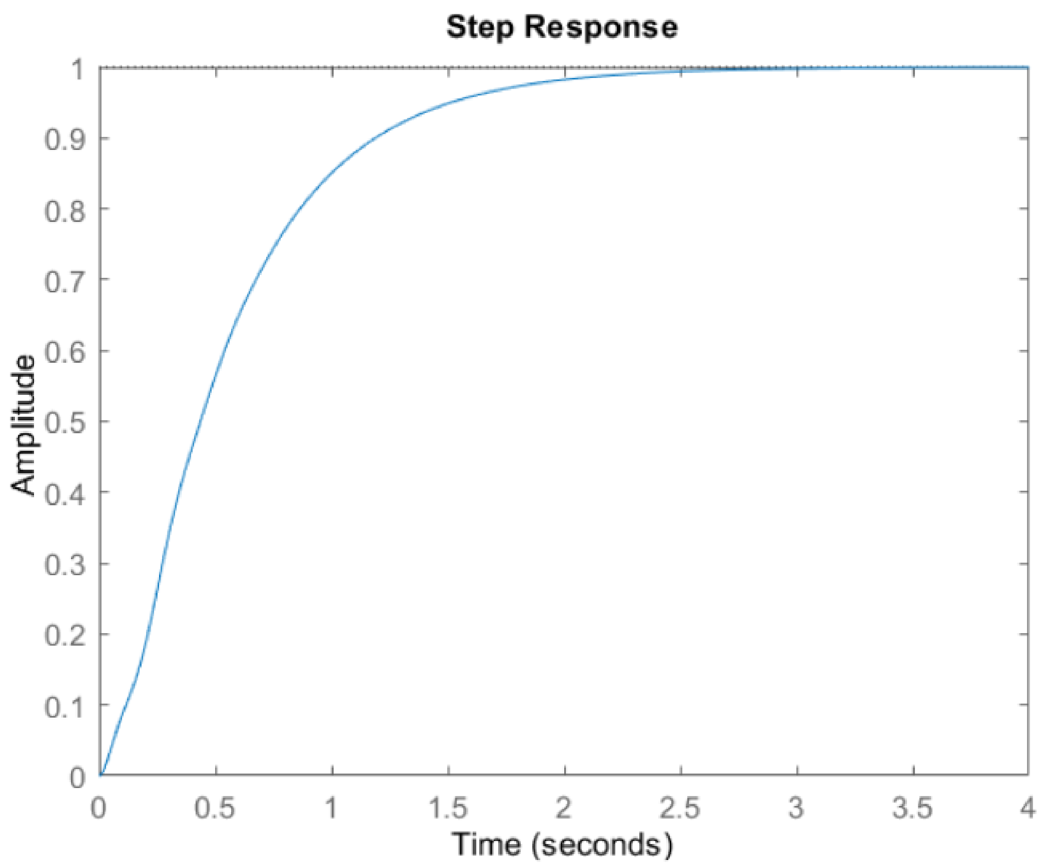
پاسخ سیستم حلقه بسته (θ و α) را به ورودی پله رسم کنید و بیشینه انحراف α، زمان نشست، زمان صعود (Rise Time) و درصد بالازدگي (Percent Of Overshoot) سیستم حلقه بسته را محاسبه نمایید.

برداشت ما این است که سیستم‌های بدون کنترل‌کننده منظور است با همین فرض آن را رسم می‌کنیم.

```
stepinfo(feedback(G_theta, 1))
```

```
ans = struct with fields:
         RiseTime: 1.0725
     SettlingTime: 1.9411
      SettlingMin: 0.9005
      SettlingMax: 0.9999
        Overshoot: 0
       Undershoot: 0
             Peak: 0.9999
         PeakTime: 4.3560
```

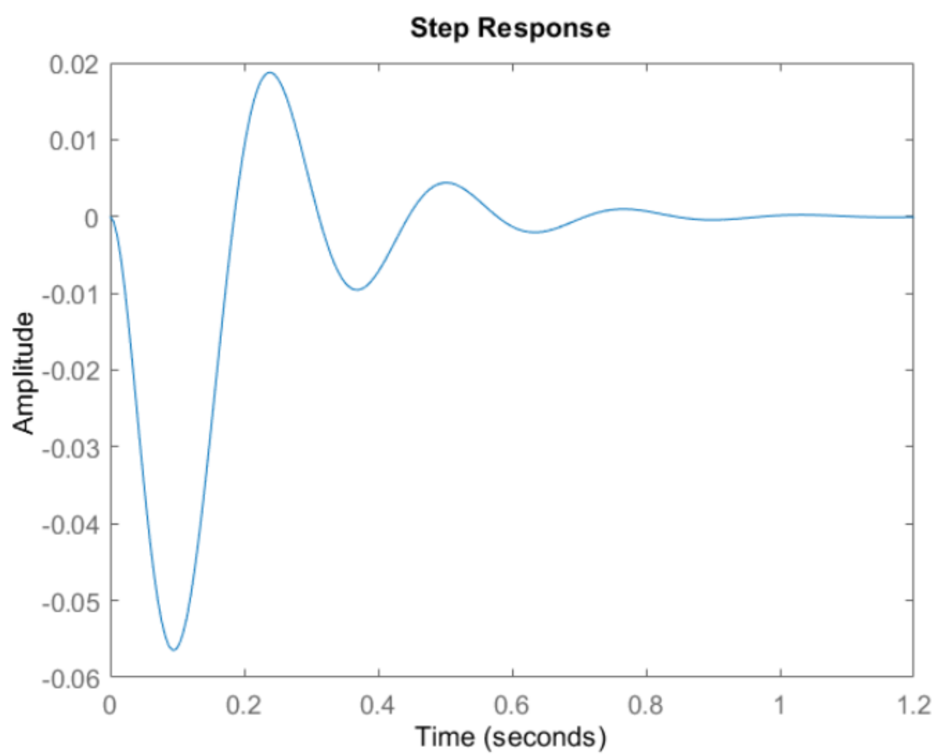
```
step(feedback(G_theta, 1))
```



```
stepinfo(feedback(G_alfa, 1))
```

```
ans = struct with fields:
        RiseTime: 1.9504e-10
    SettlingTime: 0.6784
     SettlingMin: -0.0565
     SettlingMax: 0.0187
       Overshoot: 1.7945e+11
      Undershoot: 5.9514e+10
            Peak: 0.0565
        PeakTime: 0.0931
```

```
step(feedback(G_alfa, 1))
```



+ علت تفاوت احتمالی بین متلب و عمل چیست؟

کالیبراسیون اولیه خطا دارد. استهلاک و اصطکاک بسیار موثر است.

+ مزایای کنترل‌کننده lead نسبت به PD؟

عبور مکان هندسی جبران شده از قطب‌های مطلوب، فضای آزاد بیشتر جهت طراحی

طراحی کنترل کننده‌ی Lag

طراحی

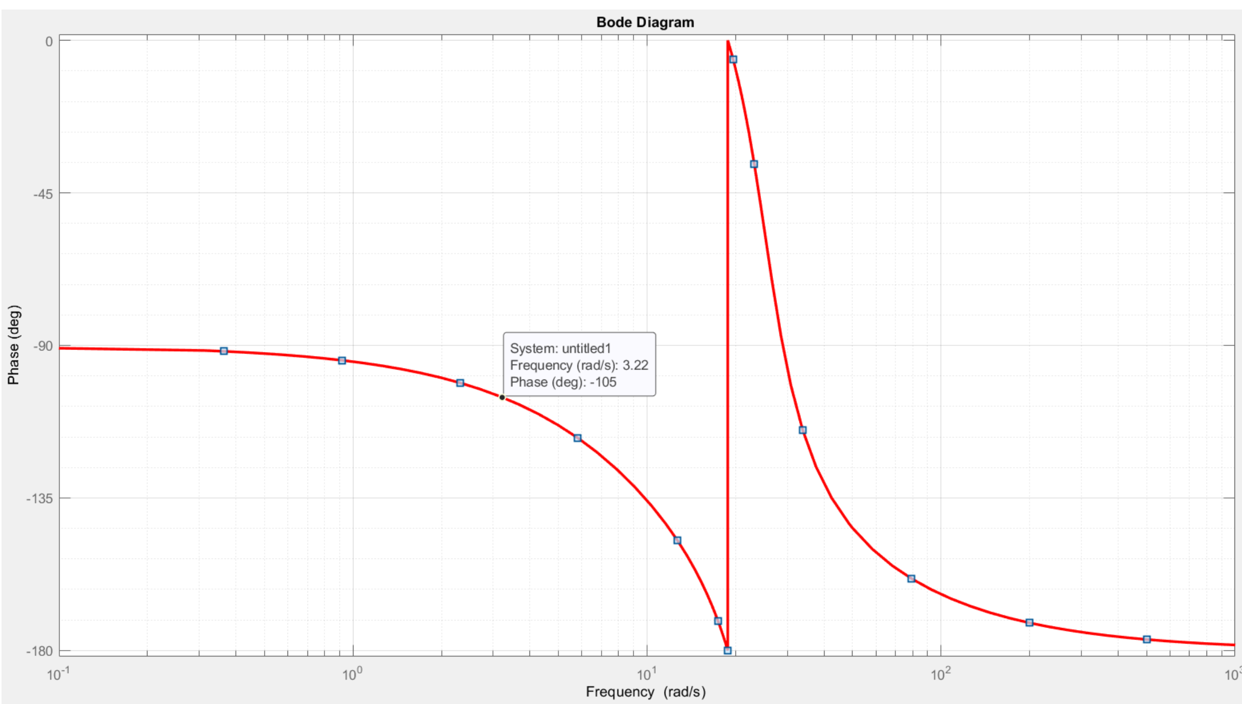
می‌دانیم که یک کنترل‌کننده‌ی پس‌فاز، به صورت زیر نمایش داده می‌شود:

$$G_C = \frac{K}{\beta} * \frac{s+z}{s+p}$$

در این کنترل‌کننده به علت رفتار پایین گذر، قطب نسبت به صفر، به محور موهومی نزدیکتر است (به بیان دیگر، صفر بسیار بزرگتر از قطب می‌باشد.). شرایط طراحی، همان شرایط کنترل‌کننده‌ی پیش‌فاز می‌باشد؛ بنابراین بهره آن نیز همان مقدار را دارد.

حال باید از روی دیاگرام Bode که از پیش ترسیم شده، حاشیه فاز را تعیین کنیم. این حاشیه فاز معادل با حاشیه فاز مطلوب به علاوه‌ی ۵ تا ۱۲ درجه است. بنابراین، خواهیم داشت:

$$\boxed{PM = 70° + 5° = 75°, \omega_c = 3.25\, rad/s}$$



$$|K \cdot G(j\omega)|_{\omega=3.25} = 9.44\ dB \rightarrow \beta = 10^{\frac{9.44}{20}} \rightarrow \boxed{\beta = 2.96}$$

برای بدست آوردن صفر و قطب باید دو نکته را مد نظر داشت:

۱- صفر را به صورتی انتخاب می‌کنیم که $z = 0.1 \cdot \omega_m$ تا اثرات پس‌فازی به حداقل برسد.

۲- ضریب β در حقیقت نسبت صفر به قطب می‌باشد.

با در نظر گرفتن دو نکته‌ی بالا، می‌توان دریافت که:

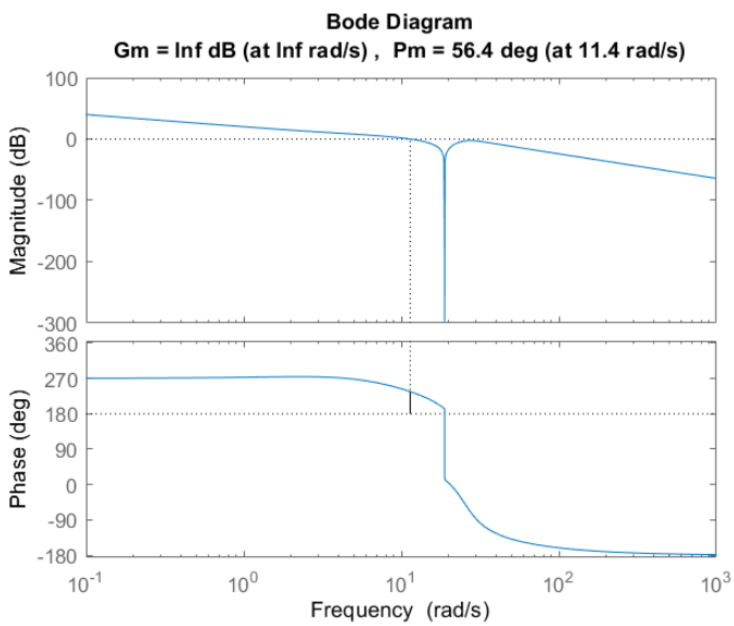
$$z = 0.325, p = 0.110$$

با قرار دادن مقادیر بدست آمده برای کنترل‌کننده‌ی پس‌فاز، خواهیم داشت:

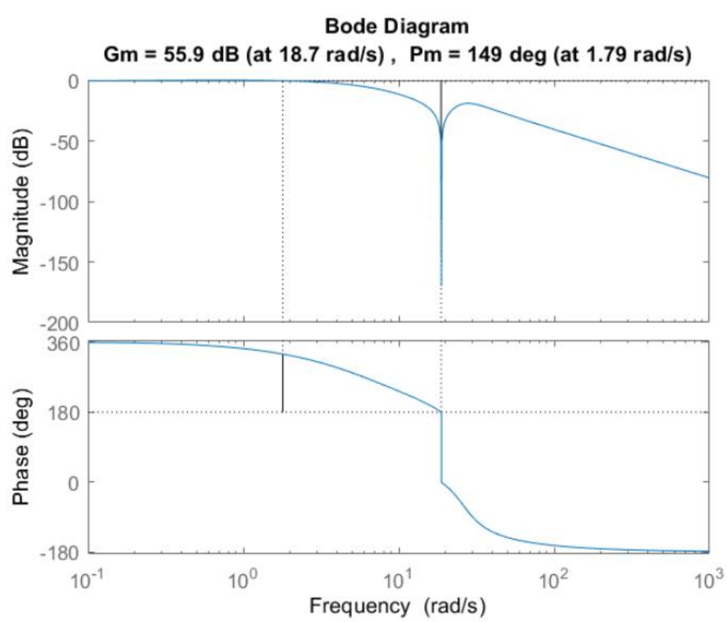
$$G = G_c * T_1 \rightarrow \boxed{G(s) = \frac{621.2s^3 + 2472s^2 + 220000s + 8753005.0}{s^5 + 37.01s^4 + 1084s^3 + 17330s^2 + 87530s}}$$

Lag Controller

```
PM_lag = 75;
w_c_lag = 3.25;
beta_lag = 2.96;
z_lag = w_c_lag * 0.1;
p_lag = z_lag / beta_lag;
C_lag = (K_ctrl / beta_lag) * ((s+z_lag)/(s+p_lag));
T_ol_lag = series(C_lag, G_theta);
margin(T_ol)
```




```
T_cl_lag = feedback(T_ol_lag, 1);
margin(T_cl_lag)
```



سوالات

نمودار پاسخ پله توابع در طراحی گذاشته شد.

طراحی کنترل‌کننده فیدبک در فضای حالت

طراحی

در کنترل‌کننده‌های فیدبک حالت، از متغیرهـای حالـت، فیـدبک گرفتـه شـده و در ضـریب K ضـرب می‌گردد و سپس به عنوان ورودی به سیستم وارد می‌شود. بنابراین خواهیم داشت:

$$\boxed{u = -Kx}$$

$$\dot{x} = Ax + Bu \rightarrow \dot{x} = Ax + B(-Kx) \rightarrow \boxed{\dot{x} = (A - BK)x}$$

پس می‌توان گفت که این بار، ماتریس گذر حالت به صورت زیر خواهد بود:

$$\Phi = \left(sI - (A - BK)\right)^{-1} \rightarrow \Phi = (sI - A + BK)^{-1}$$

حال باید دترمینان ماتریس بالا را حساب کرده و برابر با مخرج تابع تبدیلی که بر اساس شرایط مطلوب بدست می‌آید قرار داده تا ضرایب بهره‌ی K بدست آیند. این کار به علت بزرگ بودن ابعاد ماتریس گـذر حالـت بسیار دشوار بوده بنابراین از دستور زیر در محیط متلب استفاده می‌کنیم:

$$K = place(A, B, s_d)$$

در این رابطه ، s_d برداری است که شامل قطب‌های مطلوب سیستم می‌باشد. در خروجی، برداری ظاهر می‌شود که حاوی مقادیر مناسب بهره برای رسیدن به مشخصات مطلوب سیستم است.

از ما خواسته شده‌است که قطب‌های سیستم خود را در مکان‌های زیر قرار دهیم:

$$s_d = [-5.04 + j5.141 \quad -5.04 - j5.141 \quad -100 \quad -50]$$

پس از محاسبه از طریق دستور کامپیوتری خواهیم داشت:

$$\boxed{K = [14.7243 \quad -100.481 \quad 2.7367 \quad 0.849]}$$

State Space Feed-Back

```
s_d_fb = [-5.04+1j*5.141 -5.04-1j*5.141 -100 -50];
K_ssfb = place(A, B, s_d_fb);
```

```
K_ssfb = 1×4
     14.7243 -100.4809     2.7367     0.0849
```

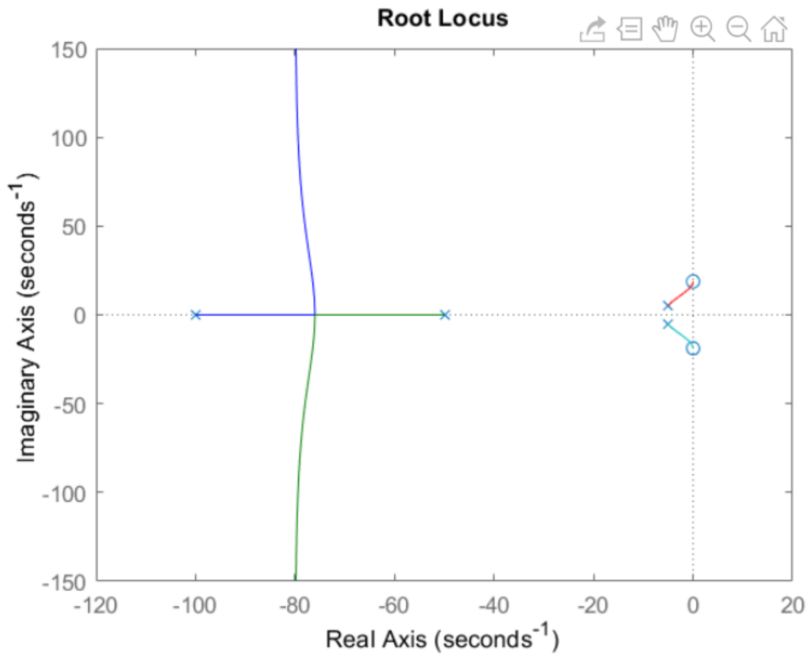
```
A_new = A - B*K_ssfb;
[N_ssfb, D_ssfb] = ss2tf(A_new, B, C, D);
G_t_ssfb = tf(N_ssfb(1,:),D_ssfb)
```

```
G_t_ssfb =

                 49.71 s^2 + 1.76e04
  ---------------------------------------------------
  s^4 + 160.1 s^3 + 6564 s^2 + 5.817e04 s + 2.592e05

Continuous-time transfer function.
```

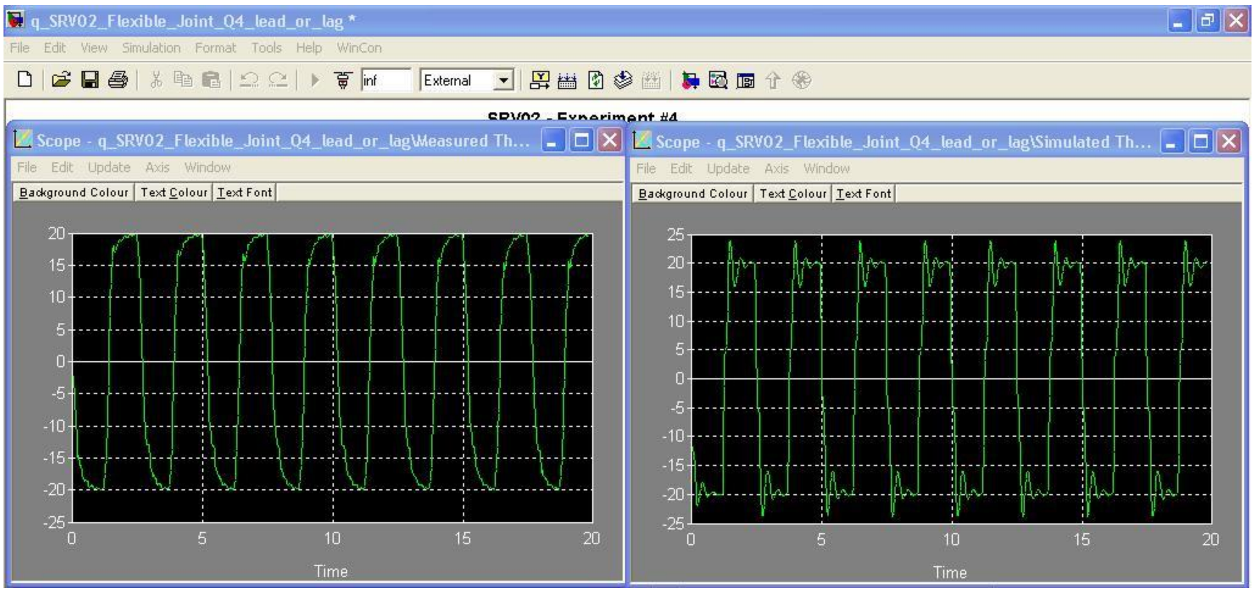
```
rlocus(G_t_ssfb)
```



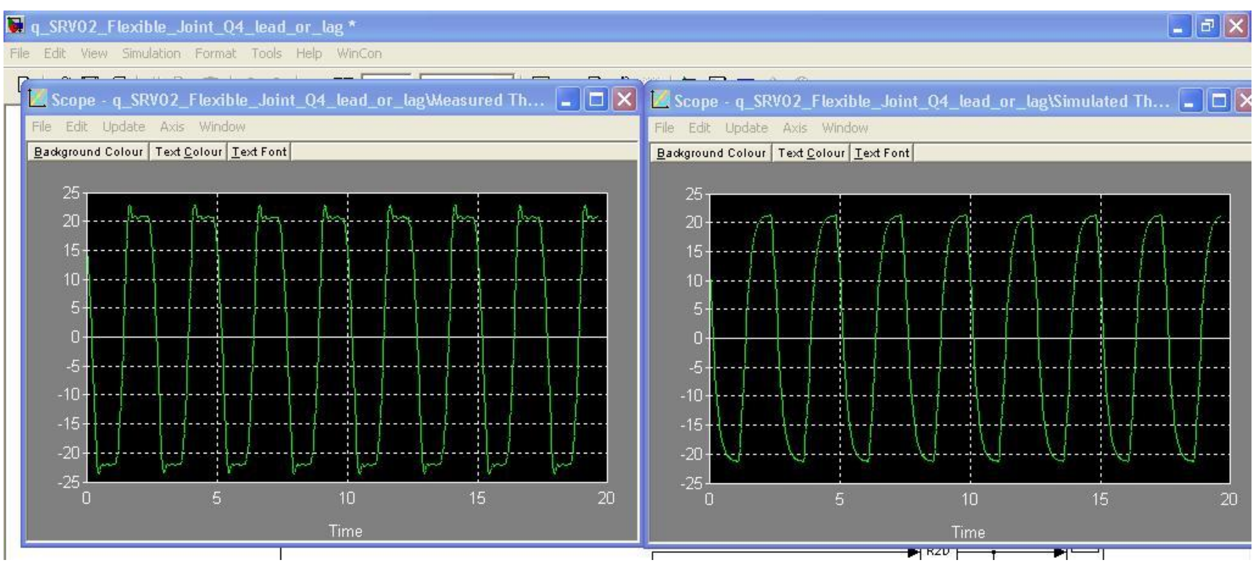
72
```
stepinfo(G_t_ssfb)
```

```
ans = struct with fields:
        RiseTime: 0.3744
    SettlingTime: 0.8688
     SettlingMin: 0.0611
     SettlingMax: 0.0706
       Overshoot: 3.9572
      Undershoot: 0
            Peak: 0.0706
        PeakTime: 0.6724
```

نتایج عملی



کنترل‌کننده Lead 1



کنترل‌کننده Lag 1