به نام خدا

دانشگاه تهران

پردیس دانشکده‌های فنی

دانشکده مهندسی برق و کامپیوتر

سیستم‌های کنترل پیشرفته

پروژه : سیستم گوی و میله

فاز اول

نورا زارعی - ۸۱۰۱۹۹۴۳۳

ثمر نیک فرجاد – ۸۱۰۱۹۹۵۰۸

فهرست

چکیده

در فاز اول پروژه قصد داریم ابتدا معادلات حالت سیستم را بدست آوریم و سپس آنها را خطی سازی کنیم و ویژگی های سیستم همچون پایداری در نقطه تعادل، کنترل پذیری و رویت پذیری و مینیمال بودن فضای حالت سیستم را بررسی کنیم. درنهایت تلاش میکنیم سیستم را با کنترلر PID کنترل کنیم.

- لازم به ذکر است که برای خوانایی بیشتر کدها، فایل متلب مجددا مرتب و بخش‌بندی شده است. بنابراین برخی کامنت‌های فایل جدید در تصاویر گزارش مشخص نیستند.

خواسته ۱

۱.۱. با توجه به متغیرهای معرفی‌شده، متغیرهای لازم را به دست می‌آوریم :

$$\left(\frac{J_b}{r^2}+m\right)\ddot{x}+(mr^2+J_b)\frac{1}{r}\ddot{\alpha}-mx\dot{\alpha}^2=mgsin(\alpha)$$

$$(mx^2+J_b+J_\omega)\ddot{\alpha}+(2m\dot{x}x+bl^2)\dot{\alpha}+Kl^2\alpha+(mr^2+J_b)\frac{1}{r}\ddot{x}-mgcos(\alpha)=u(t)lcos(\alpha)$$

$$***$$

$$\rightarrow \ddot{x}=\frac{mgsin(\alpha)+mx\dot{\alpha}^2-(mr^2+J_b)\frac{1}{r}\ddot{\alpha}}{\left(\frac{J_b}{r^2}+m\right)}$$

$$\ddot{\alpha}=\frac{u(t)lcos(\alpha)+mgcos(\alpha)-(mr^2+J_b)\frac{1}{r}\ddot{x}-Kl^2\alpha-(2m\dot{x}x+bl^2)\dot{\alpha}}{(mx^2+J_b+J_\omega)}$$

$$***$$

$$\rightarrow \ddot{\alpha}=\frac{u(t)lcos(\alpha)+mgcos(\alpha)-rmgsin(\alpha)-rmx\dot{\alpha}^2-Kl^2\alpha-(2m\dot{x}x+bl^2)\dot{\alpha}}{(mx^2+J_\omega-mr^2)}$$

$$\rightarrow \ddot{x}=\left(\frac{J_b}{r^2}+m\right)\left(mgsin(\alpha)+mx\dot{\alpha}^2-(mr^2+J_b)\frac{1}{r}\left(\frac{u(t)lcos(\alpha)+mgcos(\alpha)-rmgsin(\alpha)-rmx\dot{\alpha}^2-Kl^2\alpha-(2m\dot{x}x+bl^2)\dot{\alpha}}{(mx^2+J_\omega-mr^2)}\right)\right)$$

پس برای معادلات حالت خواهیم داشت :

$$\dot{x}_1=\dot{x}$$

$$\dot{x}_2=\ddot{x}=\left(\frac{J_b}{r^2}+m\right)\left(mgsin(x_3)+mx_1x_4^{\ 2}-\left(mr^2+J_b\right)\frac{1}{r}\left(\frac{u(t)lcos(x_3)+mgcos(x_3)-rmgsin(x_3)-rmx_1x_4^{\ 2}-Kl^2x_3-\left(2mx_2x_1+bl^2\right)x_4}{\left(mx^2+J_\omega-mr^2\right)}\right)\right)$$

$$\dot{x}_3=\dot{\alpha}$$

$$\dot{x}_4=\ddot{\alpha}=\frac{u(t)lcos(x_3)+mgcos(x_3)-rmgsin(x_3)-rmx_1x_4^{\ 2}-Kl^2x_3-\left(2mx_2x_1+bl^2\right)x_4}{\left(mx^2+J_\omega-mr^2\right)}$$

$$y_1=x_1$$

$$y_2=x_3$$

این معادلات را توسط متلب نیز به صورت زیر بدست آوردیم :

```
syms u x1 x2 x_dot_2 x3 x4 x_dot_4
% Equations
eq1 = (Jb/r^2+m)*x_dot_2 + x_dot_4*(m*r^2+Jb)/r - m*x1*x4^2 == m*g*sin(x3);
eq2 = (m*x1^2+Jb+Jw)*x_dot_4 + (2*m*x2*x1 + b*l^2)*x4 + x3*K*l^2 + x_dot_2*(m*r^2 + Jb)/r - m*g*x1*cos(x3) == u*l*cos(x3);
```

```
eqns = [eq1, eq2];
S = solve([eq1, eq2],[x_dot_2 x_dot_4]);
S.x_dot_2
```

ans =

$$\frac{144115188075855872\ (4980620899901578936320000\,x_1{}^3\,x_4{}^2 + 4885989102803448936529920\,\sin(x_3)\,x_1{}^2 + 2587952755681748866769000\,x_1\,x_4{}^2 + 2789147703944884500000\,x_2\,x_1\,x_4 - 136807694878496}{5\ (200979272962679279983055627514388099891200\,x_1{}^2 + 104317175079}$$

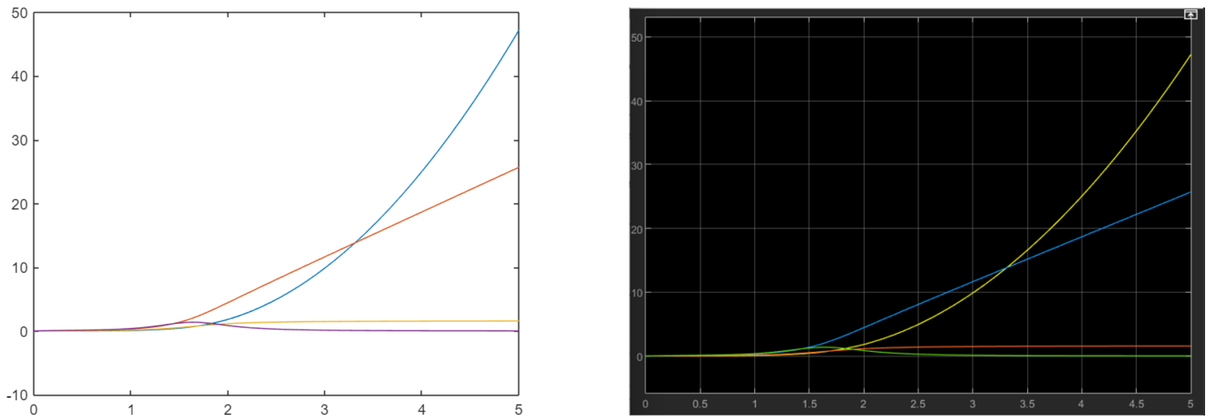
```
S.x_dot_4
```

ans =

$$-\frac{1152921504606846976\ (968857586396363856281\,6\,x_3 + 9688575863963638562816000\,x_4 + 21376202324765091363281\,25\,\sin(x_3) + 21790216437069410156250\,0\,x_1\,x_4{}^2 - 197726038040074256384000}{625\ (200979272962679279983055627514388099891200\,x_1{}^2 + 1043171750797622209218164828981602659}$$

شکل ۱ – پاسخ ساده‌شده معادلات حالت در متلب

۱.۲. نمودار متغیرهای حالت به دست آمده و فایل سیمولینک موجود را تست و بررسی کرده و هر دو را به ازای ورودی پله با اندازه ۰.۱ و شروع از لحظه ۰ و در زمان ۰ تا ۵ ثانیه رسم کردیم. همان طور که مشاهده می شود هر دو شبیه به هم هستند :



شکل ۲ – مقایسه‌ی خروجی حالات سیستم اصلی و سیستم استخراج‌شده

خواسته ۲

۲. خطی سازی حول نقطه مبدا براساس $f(x_1,x_2,x_3,x_4,u)\{at\ ss\ point\} = f(0{,}0{,}0{,}0{,}0)$ انجام خواهد شد و طبق محاسباتی که مفصل‌تر در فایل متلب قابل مشاهده است داریم :

```
%% Linearization
format long
f2_linear_to_x1(x1,x2,x3,x4,u) = diff(S.x_dot_2,x1);
f2_linear_to_x2(x1,x2,x3,x4,u) = diff(S.x_dot_2,x2);
f2_linear_to_x3(x1,x2,x3,x4,u) = diff(S.x_dot_2,x3);
f2_linear_to_x4(x1,x2,x3,x4,u) = diff(S.x_dot_2,x4);
f4_linear_to_x1(x1,x2,x3,x4,u) = diff(S.x_dot_4,x1);
f4_linear_to_x2(x1,x2,x3,x4,u) = diff(S.x_dot_4,x2);
f4_linear_to_x3(x1,x2,x3,x4,u) = diff(S.x_dot_4,x3);
f4_linear_to_x4(x1,x2,x3,x4,u) = diff(S.x_dot_4,x4);
f2_linear_to_u(x1,x2,x3,x4,u) = diff(S.x_dot_2,u);
f4_linear_to_u(x1,x2,x3,x4,u) = diff(S.x_dot_4,u);
```

$$A = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial x_3} & \frac{\partial f_1}{\partial x_4} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} & \frac{\partial f_2}{\partial x_4} \\ \frac{\partial f_3}{\partial x_1} & \frac{\partial f_3}{\partial x_2} & \frac{\partial f_3}{\partial x_3} & \frac{\partial f_3}{\partial x_4} \\ \frac{\partial f_4}{\partial x_1} & \frac{\partial f_4}{\partial x_2} & \frac{\partial f_4}{\partial x_3} & \frac{\partial f_4}{\partial x_4} \end{pmatrix} \{at\ ss\ point\} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ -0.3780 & 0 & 7.0147 & 0.0342 \\ 0 & 0 & 0 & 1 \\ 18.9001 & 0 & -0.3797 & -1.7132 \end{pmatrix}$$

$$B = \begin{pmatrix} \frac{\partial f_1}{\partial u} \\ \frac{\partial f_2}{\partial u} \\ \frac{\partial f_3}{\partial u} \\ \frac{\partial f_4}{\partial u} \end{pmatrix} \{at\ ss\ point\} = \begin{pmatrix} 0 \\ -0.0699 \\ 0 \\ 3.4964 \end{pmatrix}$$

$$C = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

$$D = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\dot{x} = Ax + Bu$$

$$y = Cx + D$$

خواسته ۳

۳. برای بررسی پایداری حول نقطه تعادل سیستم (در اینجا مبدا) مقادیر ویژه ماتریس A سیستم خطی را بدست می‌آوریم :

```
lambda = eig(A)
```

```
lambda = 4×1 complex
   2.9896 + 0.0000i
  -3.8454 + 0.0000i
  -0.4288 + 3.3669i
  -0.4288 - 3.3669i
```

شکل ۳ – مقادیر ویژه ماتریس حالت

به دلیل داشتن مقدار ویژه‌ای با مقدار حقیقی مثبت متوجه می‌شویم که سیستم خطی ناپایدار است و در نتیجه سیستم اصلی نیز ناپایدار است.

خواسته ۴

- کنترل پذیری:

ماتریس کنترل پذیری را به صورت زیر تشکیل داده و اگر رتبه کامل سطری باشد، سیستم کنترل پذیر خواهد بود.

$$C=\begin{pmatrix}B & AB & A^2B & A^3B\end{pmatrix}=\begin{pmatrix}0 & -0.0699 & 0.1198 & 24.3479\\ -0.0699 & 0.1198 & 24.3479 & -41.8051\\ 0 & 3.4965 & -5.9903 & 7.6137\\ 3.4964 & -5.9903 & 7.6137 & -8.5053\end{pmatrix}$$

```
c = [B,A*B,A^2*B,A^3*B]

c = 4x4
         0   -0.0699    0.1198   24.3479
   -0.0699    0.1198   24.3479  -41.8051
         0    3.4965   -5.9903    7.6137
    3.4965   -5.9903    7.6137   -8.5053

rank(c)

ans =
     4
```

شکل ۴ – محاسبه رنک ماتریس کنترل پذیری

از آنجا که رنک ماتریس کنترل پذیری ۴ در ۴ ما کامل و برابر با ۴ است، یعنی سطرها مستقل خطی‌اند و در نتیجه سیستم کنترل پذیر است.

- رویت پذیری:

ماتریس رویت پذیری را به صورت زیر تشکیل داده و اگر رتبه کامل ستونی باشد، سیستم رویت پذیر خواهد بود.

$$O=\begin{pmatrix}C\\ CA\\ CA^2\\ CA^3\end{pmatrix}=\begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ -0.3780 & 0 & 7.0147 & 0.0343\\ 18.9001 & 0 & -0.3797 & -1.7133\\ 0.6476 & -0.3780 & -0.0130 & 6.9560\\ -32.3809 & 18.9001 & 0.6506 & 2.5556\end{pmatrix}$$

```
o = [C;C*A;C*A^2;C*A^3]

o = 8x4
     1.0000         0         0         0
          0         0    1.0000         0
          0    1.0000         0         0
          0         0         0    1.0000
    -0.3780         0    7.0147    0.0343
    18.9001         0   -0.3797   -1.7133
     0.6476   -0.3780   -0.0130    6.9560
   -32.3809   18.9001    0.6506    2.5556

rank(o)

ans =
     4
```

شکل ۵ – محاسبه رنک ماتریس رویت پذیری

از آنجا که رنک ماتریس رویت پذیری ۸ در ۴ ما برابر با ۴ است، یعنی ستون‌ها مستقل خطی‌اند و در نتیجه سیستم رویت پذیر است.

- مینیمالیتی :

برای بررسی مینیمال بودن، دترمینان ماتریس A را محاسبه کرده و اگرغیر صفر بود سیستم مینیمال است.

```
det(A)
```

```
ans =
    -1.324358100049542e+02
```

شکل ۶ – محاسبه دترمینان ماتریس حالت

مشاهده می‌کنیم که دترمینان مخالف صفر است و سیستم کنترل پذیر و رویت پذیر است پس حذف صفر و قطب نیز نداریم و در نتیجه سیستم خود مینیمال است.

خواسته ۵

۵. برای بدست آوردن ماتریس انتقال حالت از روش فرم جردن کمک می گیریم و طبق محاسبات متلب داریم:

$$J = Q^{-1}AQ$$
$$e^{At} = Qe^{Jt}Q^{-1}$$

```
[Q,J] = eig(A)

Q = 4x4 complex
    0.1926 + 0.0000i  -0.1040 + 0.0000i  -0.0546 - 0.1400i  -0.0546 + 0.1400i
    0.5757 + 0.0000i   0.4000 + 0.0000i   0.4947 - 0.1239i   0.4947 + 0.1239i
    0.2521 + 0.0000i  -0.2292 + 0.0000i   0.0302 + 0.2375i   0.0302 - 0.2375i
    0.7536 + 0.0000i   0.8813 + 0.0000i  -0.8125 + 0.0000i  -0.8125 + 0.0000i

J = 4x4 complex
    2.9896 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
    0.0000 + 0.0000i  -3.8454 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
    0.0000 + 0.0000i   0.0000 + 0.0000i  -0.4288 + 3.3669i   0.0000 + 0.0000i
    0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i  -0.4288 - 3.3669i
```

```
e_At = Q*[exp(0.1926),0,0,0;0,exp(-3.8454),0,0;0,0,exp(-0.4288+3.3669i),0;0,0,0,exp(-0.4288-3.3669i)]*inv(Q)

e_At = 4x4 complex
    0.0006 - 0.0000i   0.1121 - 0.0000i   0.4665 + 0.0000i   0.0680 - 0.0000i
    1.2424 - 0.0000i   0.0006 + 0.0000i   0.7606 + 0.0000i   0.3539 - 0.0000i
    0.9514 - 0.0000i   0.1785 + 0.0000i   0.0005 + 0.0000i   0.0259 - 0.0000i
    0.4212 - 0.0000i   0.9514 + 0.0000i   1.2423 + 0.0000i  -0.0376 + 0.0000i
```

شکل ۷ – بدست آوردن فرم قطری، بردارهای ویژه و ماتریس انتقال حالت

خواسته ۶

تابع تبدیل را با استفاده از فرمول زیر به کمک متلب بدست می‌آوریم:

$$G(s) = C(sI - A)^{-1}B + D$$

Part 6)

```
%% Transformation Function
G = tf(ss(A,B,C,D))
```

```
G =

  From input to output...
                  -0.06993 s^2 + 24.5
   1:  ---------------------------------------------------
       s^4 + 1.713 s^3 + 0.7577 s^2 + 7.105e-15 s - 132.4

            3.496 s^2 - 1.941e-18 s - 2.913e-16
   2:  ---------------------------------------------------
       s^4 + 1.713 s^3 + 0.7577 s^2 + 7.105e-15 s - 132.4

Continuous-time transfer function.
```

شکل ۸ – بدست آوردن تابع تبدیل خطی

که با ساده سازی و حذف ترم های کوچکتر از 10^{-4} در صورت‌ها داریم:

$$G = \begin{pmatrix} \dfrac{-0.06993s^2 + 24.5}{s^4 + 1.713s^3 + 0.7577s^2 + 7.105(e-15)s - 132.4} \\ \dfrac{3.496s^2}{s^4 + 1.713s^3 + 0.7577s^2 + 7.105(e-15)s - 132.4} \end{pmatrix}$$

صفر و قطب‌های (و حتی بهره توابع تبدیل) سیستم نیز به صورت زیر خواهد بود :

```
%% System Zeros, Poles and Gain
G1_num = [0 0 -0.06993 0 24.5];
G2_num = [0 0 3.496 0 0];
G_dem = [1 1.713 0.7577 7.105e-15 -132.4];

[z1, p1, k1] = tf2zp(G1_num, G_dem)
```

```
z1 = 2×1
    18.717648098795699
   -18.717648098795703

p1 = 4×1 complex
   -3.845035030697139 + 0.000000000000000i
   -0.428698636214645 + 3.366719168838446i
   -0.428698636214645 - 3.366719168838446i
    2.989432303126429 + 0.000000000000000i

k1 =
  -0.069930000000000
```

```
[z2, p2, k2] = tf2zp(G2_num, G_dem)
```

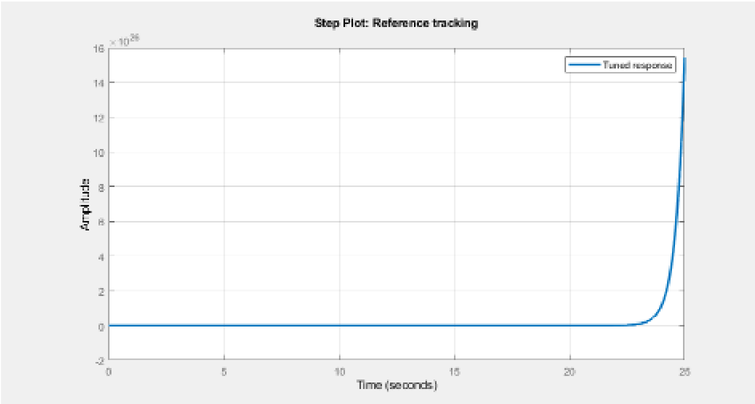
```
z2 = 2×1
         0
         0

p2 = 4×1 complex
   -3.845035030697139 + 0.000000000000000i
   -0.428698636214645 + 3.366719168838446i
   -0.428698636214645 - 3.366719168838446i
    2.989432303126429 + 0.000000000000000i

k2 =
   3.496000000000000
```

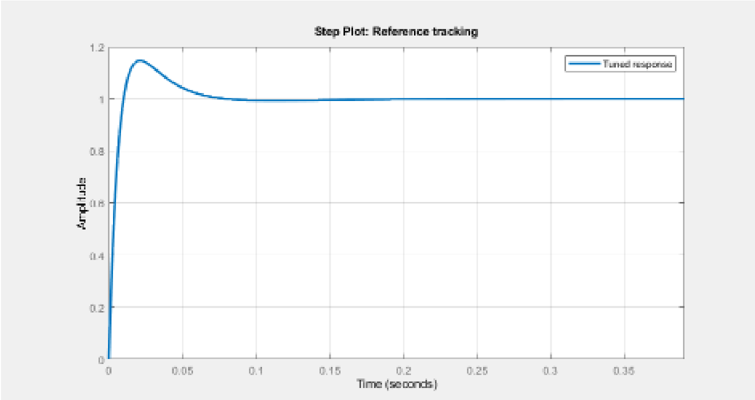
شکل ۹ – بدست آوردن بهره و صفر و قطب‌های سیستم خطی

خواسته ۷

با استفاده از ابزارهای متلب ، پاسخ خروجی‌ها بعد از اضافه شدن کنترل‌کننده‌ها را مشاهده می کنیم.

```
pidTuner(G1_approximate,'PID')
pidTuner(G2_approximate,'PID')
```





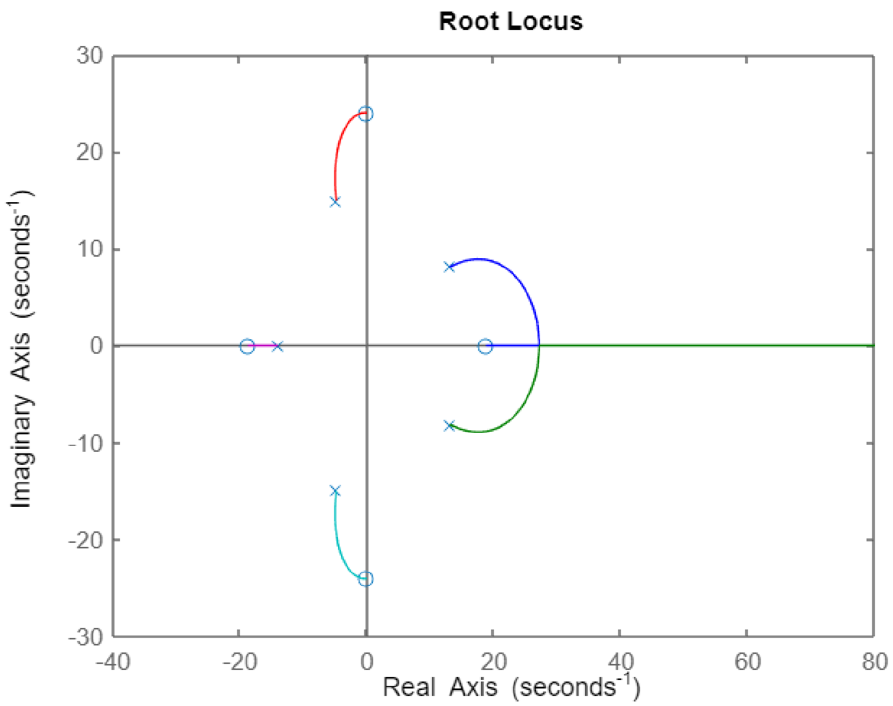
شکل ۱۰ – پاسخ خروجی‌های سیستم پس از اعمال کنترل‌کننده‌های **tune** شده

همانطور که می‌بینیم، موقعیت دارای دامنه واگراست که به معنای ناپایداری سیستم است. در تصویر زیر اطلاعات کنترل‌کننده‌های tune شده مشخص است. همچنین آنطور در اطلاعات در مقابل بخش پایداری عدد صفر (به معنای عدم پایداری) آمده است. ضرایب کنترل‌کننده‌های PID طراحی‌شده‌ی متلب نیز مشخص شده است. پس در اینجا نتوانستیم سیستم خطی را با PID کنترل نماییم که به دلیل وجود قطب ناپایدار، طبیعی بود.

```
[C_PID_1_approximate,G1_approximate_info] = pidtune(G1_approximate,'PID')
Warning: Cannot keep loop gain above 1 at low frequency and below 1 at high frequency. Try adding integrators to the loop.
C_PID_1_approximate =

             1
  Kp + Ki * --- + Kd * s
             s

  with Kp = 0.093, Ki = 0.00258, Kd = 0.837

Continuous-time PID controller in parallel form.
G1_approximate_info = struct with fields:
                Stable: 0
    CrossoverFrequency: 3.173606324483949
           PhaseMargin: NaN
[C_PID_2_approximate,G2_approximate_info] = pidtune(G2_approximate,'PID')
C_PID_2_approximate =

             1
  Kp + Ki * --- + Kd * s
             s

  with Kp = 2.77e+03, Ki = 3.33e+04, Kd = 57.7

Continuous-time PID controller in parallel form.
G2_approximate_info = struct with fields:
                Stable: 0
    CrossoverFrequency: 2.045219704308076e+02
           PhaseMargin: NaN
```

شکل ۱۱ – اطلاعات کنترل‌کننده‌های **tune** شده و وضعیت پایداری سیستم کنترل‌شده

مشاهده می شود که سیستم به ازای کنترل کننده PID قطب سمت راست محور موهومی دارد پس ناپایدار است. همچنین به ازای هیچ کنترلر PID طراحی شده سیستم پایدار نمی شود و همواره قطب سمت راست باقی می ماند:



شکل ۱۲ – مکان هندسی ریشه‌ها با کنترل کننده **PID**

خواسته 8

ابتدا دو قطب ناپایدار که دارای قسمت حقیقی مثبت هستند را حذف می کنیم. طبق محاسبات در متلب، توابع تبدیل جدید را بدست می‌آوریم :

```
p1_improved = p1(1:3);
p2_improved = p2(1:3)
```

```
p2_improved = 3×1 complex
  -3.845035030697139 + 0.000000000000000i
  -0.428698636214645 + 3.366719168838446i
  -0.428698636214645 - 3.366719168838446i
```

```
G1_improved = zpk(z1,p1_improved,k1)
```

```
G1_improved =

   -0.06993 (s-18.72) (s+18.72)
  --------------------------------
  (s+3.845) (s^2 + 0.8574s + 11.52)

Continuous-time zero/pole/gain model.
```

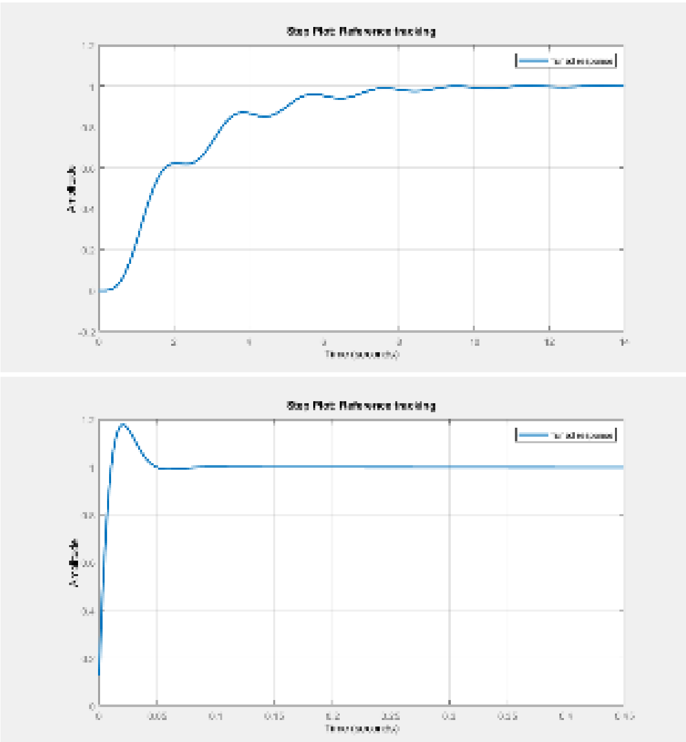
```
G2_improved = zpk(z2,p2_improved,k2)
```

```
G2_improved =

            3.496 s^2
  --------------------------------
  (s+3.845) (s^2 + 0.8574s + 11.52)

Continuous-time zero/pole/gain model.
```

شکل ۱۳ – توابع تبدیل سیستم پس از حذف قطب ناپایدار

مجددا با استفاده از ابزارهای آماده‌ی متلب، یک کنترل‌کننده طراحی می‌کنیم و پاسخ خروجی را رسم می‌کنیم.

```
pidTuner(G1_improved,'PID')
pidTuner(G2_improved,'PID')
```



شکل ۱۴ - پاسخ خروجی‌های سیستم با حذف قطب ناپایدار پس از اعمال کنترل‌کننده‌های **tune** شده

در تصویر زیر اطلاعات کنترل‌کننده‌های tune شده مشخص است.

```
[C_PID_1_improved,G1_improved_info] = pidtune(G1_improved,'PID')

C_PID_1_improved =

        1
  Ki * ---
        s

  with Ki = 0.79

Continuous-time I-only controller.
G1_improved_info = struct with fields:
                Stable: 1
    CrossoverFrequency: 0.441845787781581
           PhaseMargin: 81.528516592362379
```

```
[C_PID_2_improved,G2_improved_info] = pidtune(G2_improved,'PID')

C_PID_2_improved =

             1
  Kp + Ki * --- + Kd * s
             s

  with Kp = 48.4, Ki = 4.02e+03, Kd = 0.0401

Continuous-time PID controller in parallel form.
G2_improved_info = struct with fields:
                Stable: 1
    CrossoverFrequency: 1.777615127234068e+02
           PhaseMargin: 73.789642253631811
```

شکل ۱۴ – اطلاعات کنترل‌کننده‌های **tune** شده و وضعیت پایداری سیستم کنترل‌شده با حذف قطب ناپایدار

با توجه به سوالی که از یکی از دستیاران کردیم، هدف این فاز کنترل موقعیت است و لذا از اینجا به بعد ما بر روی کنترل خروجی موقعیت گوی تمرکز می‌کنیم. همانطور که می‌بینیم، بعد از حذف قطب و طراحی کنترل‌کننده، خروجی موقعیت گوی به مقدار نهایی همگرا شده و کنترل می‌شود. در قسمت اطلاعات نیز پایداری عدد یک را دارد. کنترل‌کننده‌ی طراحی شده شامل یک انتگرال‌گیر با ضریب ۰.۷۹ است.

با توجه به تابع تبدیل بدست آمده برای خروجی موقعیت پس از حذف قطب ناپایدار، یک تحقق خطی از آن را بدست می‌آوریم :

```
G1_improved =

    -0.06993 (s-18.72) (s+18.72)
  --------------------------------
  (s+3.845) (s^2 + 0.8574s + 11.52)
```

$$A = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -44.2944 & -14.816703 & -4.7024 \end{pmatrix}$$

$$B = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

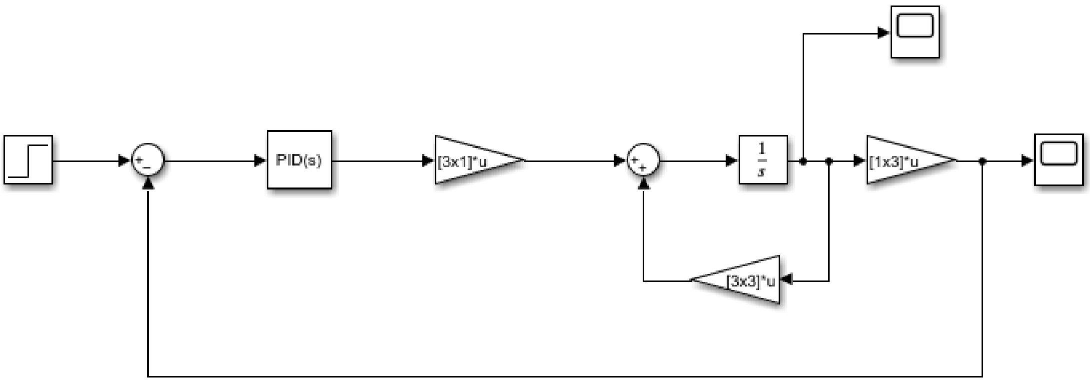
$$C = \begin{pmatrix} 24.506157312 & 0 & -0.069993 \end{pmatrix}$$

$$D = 0$$

$$\dot{x} = Ax + Bu$$

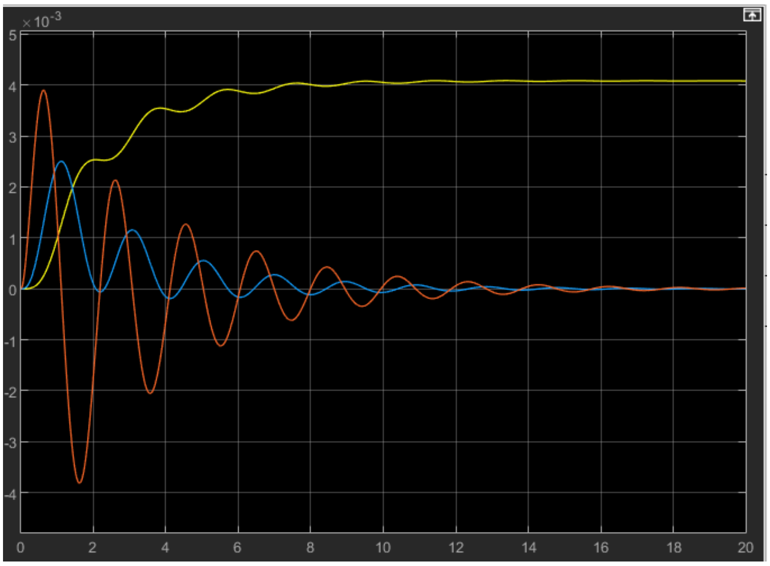
$$y = Cx + D$$

ما این تحقق را به صورت بلوکی در سیمولینک پیاده کرده و سپس ضرایب بدست آمده برای کنترل‌کننده را به آن اعمال کردیم که بلوک دیاگرام آن در شکل زیر قابل مشاهده است :



شکل ۱۶ – بلوک دیاگرام سیستم جبران‌شده خطی در سیمولینک

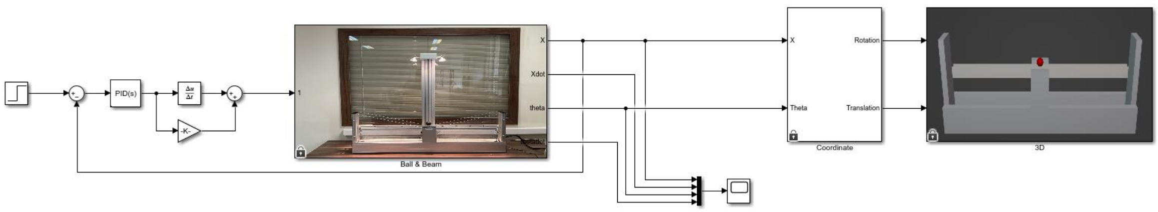
پاسخی که از متغیرهای حالت گرفتیم به شکل زیر بود :



شکل ۱۷ – پاسخ متغیرهای حالت سیستم جبران‌شده خطی در سیمولینک

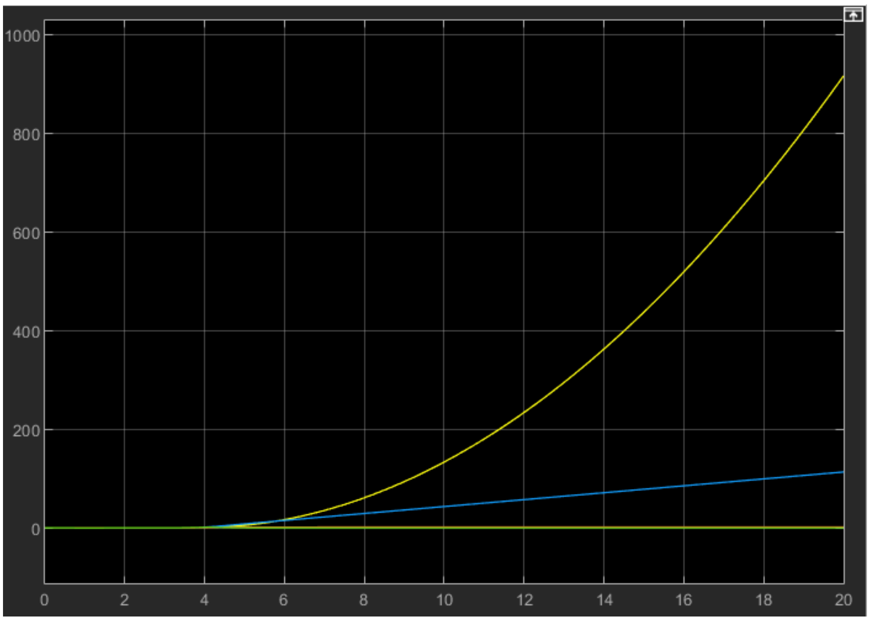
همانطور که می‌بینیم متغیرهای حالت همگی به مقداری همگرا شده و نوساناتشان نیز میرا شدند، که این به معنای پایداری سیستم می‌باشد. پس توانستیم سیستم خطی پس از حذف قطب را با استفاده از PID کنترل کنیم که انتظار آن را نیز داشتیم.

همچنین در مورد سیستم غیرخطی نیز ما کنترل‌کننده را به آن اعمال کردیم و صفری در همانجا که قطب ناپایدار داشتیم ضرب کرده، که بلوک دیاگرام آن در زیر مشخص است :



شکل ۱۸ – بلوک دیاگرام سیستم جبران‌شده غیرخطی در سیمولینک

پاسخی که از متغیرهای حالت گرفتیم به شکل زیر بود :



شکل ۱۹ - پاسخ متغیرهای حالت سیستم جبران‌شده غیرخطی در سیمولینک

همانطور که می‌بینیم برخی از متغیرهای حالت واگرا می‌شوند، که این به معنای ناپایداری سیستم می‌باشد. پس در اینجا باز هم نتوانستیم سیستم غیرخطی پس از حذف قطب را با استفاده از PID کنترل کنیم. (به دلیل بالا بردن زمان شبیه‌سازی برای نمایش بهتر متغیرهای حالت سیستم جبران‌شده‌ی خطی، اینجا نمودار قرمز متغیر حالت چهارم به خوبی دیده نمی شود.)

- کنترل پذیری:

ماتریس کنترل پذیری را به صورت زیر تشکیل داده و اگر رتبه کامل سطری باشد، سیستم کنترل پذیر خواهد بود.

$$C = (B \quad AB \quad A^2B) = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & -4.7024 \\ 1 & -4.7024 & 7.2959 \end{pmatrix}$$

```
c_improved = [B_improved, A_improved*B_improved, A_improved^2*B_improved]
```

```
c_improved = 3×3
         0         0    1.0000
         0    1.0000   -4.7024
    1.0000   -4.7024    7.2959
```

شکل ۲۰ – محاسبه رنک ماتریس کنترل پذیری

از آنجا که رنک ماتریس کنترل پذیری ۳ در ۳ ما کامل و برابر با ۳ است، یعنی سطرها مستقل خطی‌اند و در نتیجه سیستم کنترل پذیر است.

- رویت پذیری:

ماتریس رویت پذیری را به صورت زیر تشکیل داده و اگر رتبه کامل ستونی باشد، سیستم رویت پذیر خواهد بود.

$$O = \begin{pmatrix} C \\ CA \\ CA^2 \end{pmatrix} = \begin{pmatrix} 24.5062 & 0 & -0.0699 \\ 3.0834 & 25.5423 & 0.3288 \\ -14.4993 & -1.7889 & 23.9960 \end{pmatrix}$$

```
o_improved = [C_improved; C_improved*A_improved; C_improved*A_improved^2]
```

```
o_improved = 3×3
   24.5062         0   -0.0699
    3.0834   25.5423    0.3288
  -14.4993   -1.7889   23.9960
```

شکل ۲۱ – محاسبه رنک ماتریس رویت پذیری

از آنجا که رنک ماتریس رویت پذیری ۳ در ۳ ما کامل و برابر با ۳ است، یعنی ستون‌ها مستقل خطی‌اند و در نتیجه سیستم رویت پذیر است.