



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

پروژه کنترل مدرن

تحلیل سیستم ... و طراحی کنترل کننده فیدبک حالت و رؤیت گر برای
آن با رویکرد کنترل مدرن

نگارش

امیرحسین دژدار-9923026

مهشید جعفری-9923015

محمد امین محتشمی-9923120

زینب سادات میرهاشمی-9923125

نگار تهلوی-9923102

استاد

دکتر هاجر عطریانفر

پاییز 1402

چکیده

در این گزارش ابتدا به مدلسازی سیستم پاندول معکوس پرداخته می‌شود سپس مدل غیرخطی بدست آمده خطی شده و با استفاده از ابزارهای کنترل مدرن، پایداری، رؤیت‌پذیری و کنترل‌پذیری وسیله بررسی می‌شود. در ادامه برای این وسیله کنترل‌کننده فیدبک حالت، فیدبک حالت با پیش‌جبران‌ساز استاتیکی و دینامیکی (انتگرالی) طراحی می‌شود. در انتها نیز رؤیتگرهای مرتبه کامل و مرتبه کاهش یافته طراحی شده و نتایج عملکرد آن‌ها به صورت شبیه‌سازی صحنه‌گذاری می‌شود.

واژه‌های کلیدی:

صفحه

فهرست مطالب

1.....	مقدمه
2.....	معرفی سیستم
3.....	فاز اول
4.....	مدلسازی سیستم و معادلات غیرخطی حاکم بر سیستم
7.....	پارامترهای سیستم
8.....	خطی سازی مدل غیرخطی و استخراج مدل فضای حالت
8.....	سوال 2
13.....	سوال 3
14.....	سوال 4
16.....	سوال 6
17.....	سوال 7
18.....	سوال 8
19.....	سوال 10
21.....	سوال 11
22.....	فاز دوم
23.....	سوال 1
24.....	سوال 2
25.....	سوال 3
28.....	سوال 4
30.....	سوال 5
30.....	سوال 6

مقدمه

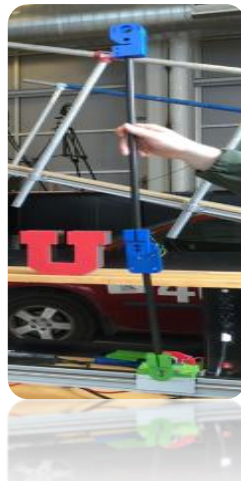
کنترل موقعیت یکی از مسائل مطرح در علم کنترل می باشد. تجزیه و تحلیل و کنترل مدل فیزیکی ربات (پلنت) یکی از موضوعات اصلی و مهم در زمینه کنترل می باشد.

در این پروژه، هدف ما ابتدا تحلیل و خطی سازی پاندول معکوس دو لینکه می باشد. سیستم پاندول دوگانه به عنوان مدل فیزیکی بازوی ربات انعطاف پذیر در این تحقیق مورد توجه قرار گرفته و رفتار آن در مورد یک حالت عملیاتی شامل حالت های سیستم بررسی شده است. طراحی یک طرح فیدبک تلاش می کند تا این حالت ها را در موقعیت مورد نظر به گونه ای اختصاص دهد که موقعیت بازو به طور مطلوب کنترل شود. لازم به ذکر است که سرعت حرکت بالای بازو به معنی تغییرات زیادی از نیروهای اینرسی وارد بر بازو است پس از آن پیاده سازی موارد مطرح شده در کنترل مدرن می باشد.

معرفی سیستم

پاندول معکوس دو لینکه یکی از مسائل قدیمی فیلد کنترل و دینامیک می باشد و عموماً برای آزمایش و پیاده سازی الگوریتم های کنترلی استفاده می شود.

این سیستم شامل آونگی است که آونگ دیگری به انتهای آن متصل است و یک سیستم فیزیکی ساده را تشکیل میدهد، مشخص است که این سیستم به شدت حساس به حالات اولیه می باشد و سیستم به شدت آشفته می باشد.



شکل 0-1 نمونه ای از یک پاندول معکوس دو لینکه

فاز اول پروژه

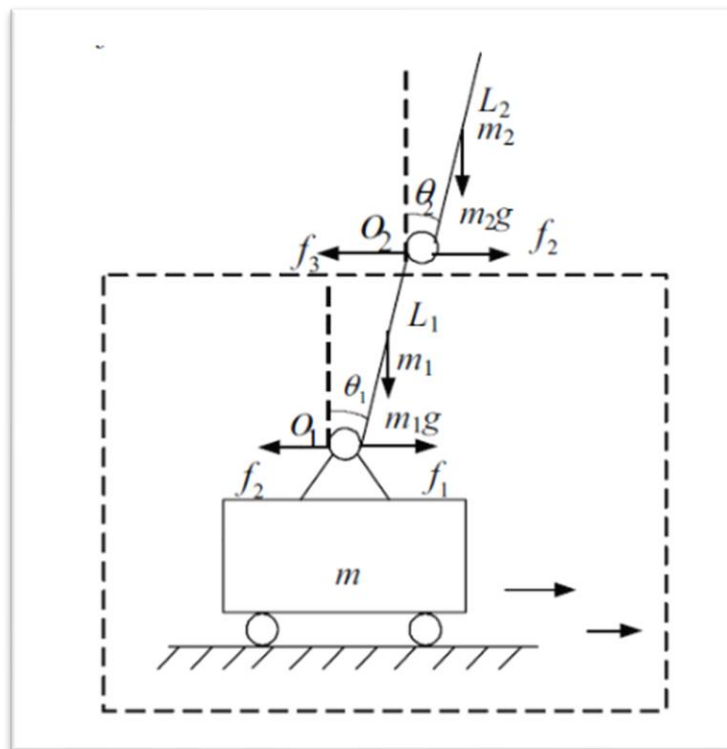
**مدلسازی سیستم، بررسی پایداری، کنترل پذیری و رؤیت پذیری
آن**

مدلسازی سیستم

فرضیات مورد استفاده در مدلسازی حرکت پاندول معکوس دو لینکه به صورت زیر می باشد.

- پاندول معکوس دو لینکه یک جسم صلب در نظر گرفته می شود.
- از اثرات زمین صرف نظر شده و زمین تخت (مسطح) فرض می شود.
- ساختار کوادروتور متقارن است.
- مرکز ثقل^۱، مرکز هندسی، مرکز تراست^۲ و مبدأ دستگاه بدنی کوادروتور بر هم منطبق هستند.
- جرم و ممان های اینرسی ثابت اند.

معادلات غیرخطی حاکم بر سیستم



شکل 1-2 مدل پاندول معکوس دو لینکه

¹ Center of Gravity (CoG)

² Center of Thrust (CoT)

$$T_0 = \frac{1}{2} m_x \dot{x}^2 \quad (1-2)$$

$$V_0 = 0 \quad (2-2)$$

$$T_1 = \frac{1}{2} J_1 \dot{\theta}_1^2 + \frac{1}{2} m_1 \left\{ \left[\frac{d}{dx} (x + l_1 \sin \theta_1) \right]^2 + \left[\frac{d}{dx} (l_1 \cos \theta_1) \right]^2 \right\} \quad (3-2)$$

$$V_1 = m_1 g l_1 \cos \theta_1 \quad (4-2)$$

$$T_2 = \frac{1}{2} J_2 \dot{\theta}_2^2 + \frac{1}{2} m_2 \left\{ \left[\frac{d}{dx} (x + l_1 \sin \theta_1 + l_2 \sin \theta_2) \right]^2 + \left[\frac{d}{dx} (l_1 \cos \theta_1 + l_2 \cos \theta_2) \right]^2 \right\} \quad (5-2)$$

$$V_2 = m_2 g (l_1 \cos \theta_1 + l_2 \cos \theta_2) \quad (6-2)$$

$$\begin{aligned} T_{total} &= T_0 + T_1 + T_2 \quad (7-2) \\ &= \frac{1}{2} m_x \dot{x}^2 + \frac{1}{2} J_1 \dot{\theta}_1^2 \\ &\quad + \frac{1}{2} m_1 \left\{ \left[\frac{d}{dx} (x + l_1 \sin \theta_1) \right]^2 + \left[\frac{d}{dx} (l_1 \cos \theta_1) \right]^2 \right\} \\ &\quad + \frac{1}{2} J_2 \dot{\theta}_2^2 \\ &\quad + \frac{1}{2} m_2 \left\{ \left[\frac{d}{dx} (x + l_1 \sin \theta_1 + l_2 \sin \theta_2) \right]^2 + \left[\frac{d}{dx} (l_1 \cos \theta_1 + l_2 \cos \theta_2) \right]^2 \right\} \end{aligned}$$

$$V_{total} = V_0 + V_1 + V_2 m_1 g l_1 \cos \theta_1 + m_2 g (l_1 \cos \theta_1 + l_2 \cos \theta_2) \quad (8-2)$$

$$L = L(x, \theta_1, \theta_2, \dot{x}, \dot{\theta}_1, \dot{\theta}_2) = T - V \quad (9-2)$$

$$H_1(z)\ddot{z} = H_2(z, \dot{z})\dot{z} + h_2(z) + h_0 u \quad (10-2)$$

$$z = (x, \theta_1, \theta_2)^T \quad (11-2)$$

$$h_0 = [1, 0, 0]^T \quad (12-2)$$

$$H_1(z) = \begin{bmatrix} a_0 & a_1 \cos \theta_1 & a_2 \cos \theta_2 \\ a_1 \cos \theta_1 & b_1 & a_2 l_1 \cos \theta_2 - \theta_1 \\ a_2 \cos \theta_2 & a_2 l_1 \cos \theta_2 - \theta_1 & b_2 \end{bmatrix} \quad (13-2)$$

$$H_2(z) = \begin{bmatrix} -f_0 & a_1 \sin \theta_1 \cdot \dot{\theta}_1 & a_2 \sin \theta_2 \dot{\theta}_2 \\ 0 & -f_1 - f_2 & a_2 l_1 \sin \theta_2 \dot{\theta}_2 \\ 0 & -a_2 l_1 \sin \theta_2 - \theta_1 \dot{\theta}_1 + f_2 & -f_2 - f_3 \end{bmatrix} \quad (14-2)$$

$$h_3(z) = [0, a_1 g \sin \theta_1, a_2 g \sin \theta_2]^T \quad (15-2)$$

$$a_0 = m_c + m_1 + m_2 \quad (16-2)$$

$$a_1 = m_1 l_1 + m_2 l_2 \quad (17-2)$$

$$a_2 = m_2 l_2 \quad (18-2)$$

$$b_1 = J_1 + m_1 l_1^2 + m_2 l_2^2 \quad (19-2)$$

$$b_2 = J_2 + m_2 l_2^2 \quad (20-2)$$

$$X = (x, \theta_1, \theta_2, \dot{x}, \dot{\theta}_1, \dot{\theta}_2)^T \quad (21-2)$$

$$\dot{X} = \begin{bmatrix} 0 & I_{4 \times 4} \\ E^{-1}H & E^{-1}H \end{bmatrix} X + \begin{bmatrix} 0 \\ E^{-1}h_0 \end{bmatrix} U = AX + BU \quad (22-2)$$

$$Y = CX \quad (23-2)$$

$$E = \begin{bmatrix} a_0 & a_1 & a_2 \\ a_1 & b_1 & a_2 l_1 \\ a_2 & a_2 l_1 & b_2 \end{bmatrix} \quad (24-2)$$

$$H = \begin{bmatrix} 0 & 0 & 0 \\ 0 & a_1 g & 0 \\ 0 & 0 & a_2 g \end{bmatrix} \quad (25-2)$$

$$G = \begin{bmatrix} f_0 & 0 & 0 \\ 0 & -f_1 - f_2 & f_2 \\ 0 & f_2 & -f_2 \end{bmatrix}^T \quad (26-2)$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \quad (27-2)$$

1-1-2- پارامترهای سیستم

پارامترهای مورد نیاز جهت مدلسازی کوادروتور به صورت زیر می باشد.

جدول 1-2- پارامترهای مورد نیاز برای مدلسازی پاندول معکوس دو لینکه

اندازه	واحد (در بردارها برای مؤلفه ها)	نمایش (نماد)	پارامتر
1.5	Kg	m_c	وزن گاری
0.04064	m	l	طول لینک ۱
0.4	Kg	m	جرم لینک ۱
0.6878	m	l	طول لینک ۲

0.22	Kg	m	جرم لینک ۲
0.5	N	f	اصطکاک 0
0.2	N	f	اصطکاک 1
0.2	N	f	اصطکاک 2
0.0407	—	—	J_1
0.359	—	—	J_2

خطی سازی مدل غیر خطی و استخراج مدل فضای حالت

برای خطی سازی سیستم ابتدا باید معادلات دینامیکی سیستم را بدست آوریم. برای این کار از معادلات لاگرانژی استفاده میکنیم.

۲ متغیر حالت ما در مسئله پاندول ۲ لینکه، زاویه لولا و زاویه پاندول پایین هستند که آنها را بصورت تتا ۱ و تتا ۲ در نظر میگیریم.

سوال (2)

برای تعریف معادله دیفرانسیل سیستم در متلب ابتدا تمام متغیرها و خود سیستم را تعریف میکنیم که آن قسمت از کد بصورت کامل در ادامه آورده شده است :

% Masses of the first and second pendulum links.

M1 = 0.2;

m2 = 0.2;

% Lengths of the first and second pendulum links.

R1=0.2;

r2=0.2;

% Initial angles of the first and second pendulum links.

Theta10 = 0;

theta20 = 0;

% Spring constants for rotational springs.

```

Ktheta1 = 1;
ktheta2 = 1;
% Damping coefficients for rotational damping.
Dtheta1 = 0.06;
Dtheta2 = 0.06;
% Moments of inertia for the first and second pendulum links.
Jtheta1 = 1;
jtheta2 = 1;
% Acceleration due to gravity.
G = 9.8;

```

```

syms theta1 theta1_d theta1_dd theta2...
    theta2_d theta2_dd u ...
    x1 x1_d x2 x2_d x3 x3_d x4 x4_d s

```

مقادیر موجود در سیستم همگی از جنس طول و مشتقات دوم زوایا میباشند که تتا ۱ و تتا ۲ از روی آنها بدست آورده میشوند.

معادلات دیفرانسیلی سیستم:

```

eqns1 = [((m1+m2)*r1^2+jtheta1)*theta1_dd+m2*r1*r2*theta2_dd*(cos(theta1-
theta2))+m2*r1*r2*theta2_d^2*(sin(theta1-theta2))+(theta1-
theta10)/ktheta1+Dtheta1*theta1_d+(m1+m2)*g*r1*sin(theta1)== 0
(m2*r2^2+jtheta2)*theta2_dd+m2*r1*r2*theta1_dd*cos(theta1-theta2)-
m2*r1*r2*theta1_d^2*sin(theta1-theta2)+(theta2-
theta20)/ktheta2+Dtheta2*theta2_d+m2*g*r2*sin(theta2)== 0];

```

حال از تابع solve متلب برای بدست آوردن مقادیر مورد نیاز استفاده میکنیم:

```

S1 = solve(eqns1,[theta1_dd theta2_dd]);

```

سپس به درایه اول هر کدام، مقادیر اولیه را به آنها اضافه میکنیم :

```

theta1_dd=S1.theta1_dd-16.01*u;
theta2_dd=S1.theta2_dd+38.84*u;

```

حال با اختیار داشتن معادلات سیستم، سیستم را حول نقطه تعادل خطی سازی میکنیم:

با توجه به شکل فیزیکی پاندول ها درمیابیم که این سیستم ۲ نقطه تعادل دارد. یکی در زاویه ۰ (تعادل ناپایدار) و (تعادل ناپایدار). بنابراین خطی سازی را حول هر دو انجام میدهیم :

حول ۰ درجه :

```
%% Linearization around theta1 , theta2 = 0
```

```
X = [theta1 theta1_d theta2 theta2_d];
```

```
f = [theta1_d;theta1_dd;theta2_d;theta2_dd];
```

```
As = jacobian(f,X);
```

```
A = subs(As, [X,u],[0.000001 0 0 0 0]);
```

```
A1 = double(A);
```

```
A1
```

```
Bs = jacobian(f,u);
```

```
B = subs(Bs, [X,u],[0.000001 0 0 0 0]);
```

```
B1 = double(B);
```

```
B1
```

```
C = [0 0 1 0];
```

```
D = 0;
```

```
eigen_vals1 = eig(A1);
```

```
[V1 , J1] = ordan(A1);
```

```
[V1 , J1] = cdf2rdf(V1,J1);
```

```
V1
```

```
Aj = inv(V1)*A1*V1;
```

```
Bj = inv(V1)*B1;
```

```
Cj = C*V1;
```

```
Aj
```

```
sys1 = ss(A1,B1,C,D);
```

```
g1 = tf(sys1);
```

```
g1
```

```
% figure(1)
```

```
% rlocus(g1);
```

```
phi_s1 = inv(s*eye(4)-A1);
```

```
phi_s1 = simplify(phi_s1);
```

```
phi_s1
```

حول ۱۸۰ درجه :

```
%% Linearization around theta1 , theta2 = pi
```

```

X = [theta1 theta1_d theta2 theta2_d];

f = [theta1_d;theta1_dd;theta2_d;theta2_dd];

As = jacobian(f,X);
A = subs(As, [X,u],[pi+0.000001 0 pi 0 0]);
A2 = double(A);
A2
Bs = jacobian(f,u);
B = subs(Bs, [X,u],[pi+0.000001 0 pi 0 0]);
B2 = double(B);
B2
C = [0 0 1 0];

D = 0;

eigen_vals2 = eig(A2);

[V2 , J2] = ordan(A2);
[V2 , J2] = cdf2rdf(V2,J2);
V2
Ajpi = inv(V2)*A2*V2;
Bjpi = inv(V2)*B2;
Cjpi = C*V2;
Ajpi
sys2 = ss(A2,B2,C,D);
g2 = tf(sys2);
g2
% figure(1)
% rlocus(g2);

phi_s2 = inv(s*eye(4)-A2);
phi_s2 = simplify(phi_s2);
phi_s2

```

A1 و A2 همان معادلات خطی شده سیستم حول نقاط تعادل هستند (با استفاده از تابع $(jacobian(f,X))$:

A1 =

0	1.0000	0	0
-1.756	-0.0591	0.0109	0.0005
0	0	0	1.0000
0.0139	0.0005	-1.3810	-0.0595

B1=

0

-16.0100

0

38.8400

A2=

0	1.0000	0	0
-0.2126	-0.0591	0.0047	0.0005
0	0	0	1.0000
0.0017	0.0005	-0.6032	-0.0595

B2=

0

-16.0100

0

38.8400

در کدهای بالا (بخش خطی سازی) با استفاده از ماتریکس ژاکوبین f, x ماتریس A و با استفاده از ماتریکس ژاکوبین f, u ماتریس B را بدست می آوریم. (در بالا آمده)
 باتوجه به معادلات سیستم ماتریس های c, d هم بصورت زیر هستند:

$C = [0 \ 0 \ 1 \ 0];$

$D = 0;$

سوال ۳)

برای بدست آوردن بلوک جردن از فانکشن آماده متلب استفاده میکنیم. سپس برای تبدیل فرم جردن مختلط به بلوک های حقیقی از دستور `cdf2rdf` استفاده میکنیم :

```
[V1 , J1] = ordan(A1);  
[V1 , J1] = cdf2rdf(V1,J1);
```

خروجی :

برای نقطه تعادل 0 درجه :

V1=

0.0007	-0.0246	1.3894	20.2767
0.0289	0.0015	-26.907	1.2418
-0.0215	-0.8505	-0.0168	-0.7544
1.0000	0	1.0000	0

Aj =

-0.0297	1.1746	0	0
-1.1746	-0.0297	0	0
0	0	-0.0295	1.3250
0	0	-1.3250	-0.0295

برای نقطه تعادل ۱۸۰ درجه :

V2=

-0.0006	0.0156	-95.6494	-492.8656
-0.0121	-0.0009	229.6064	-29.459

-0.0493	-1.2866	-0.1389	-2.1644
1.0000	0	1.0000	0

Ajpi=

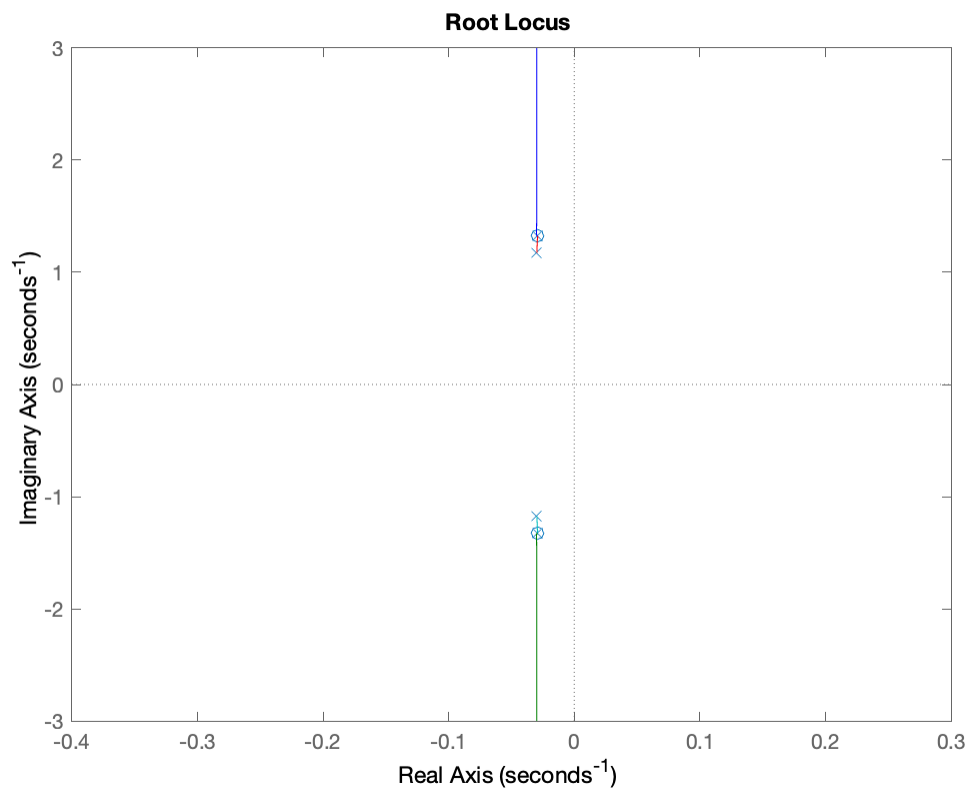
-0.0298	0.7761	0	0
-0.7761	-0.0298	0	0
0	0	-0.0295	0.4601
0	0	-0.4601	-0.0295

سوال (۴)

با استفاده از کد های متلب زیر این کار را انجام می‌دهیم:

```
sys1 = ss(A1,B1,C,D);
g1 = tf(sys1);
g1
figure(1)
rlocus(g1);
```

$$g1 = \frac{38.84 s^2 + 2.286 s + 67.98}{s^4 + 0.1186 s^3 + 3.141 s^2 + 0.1861 s + 2.425}$$

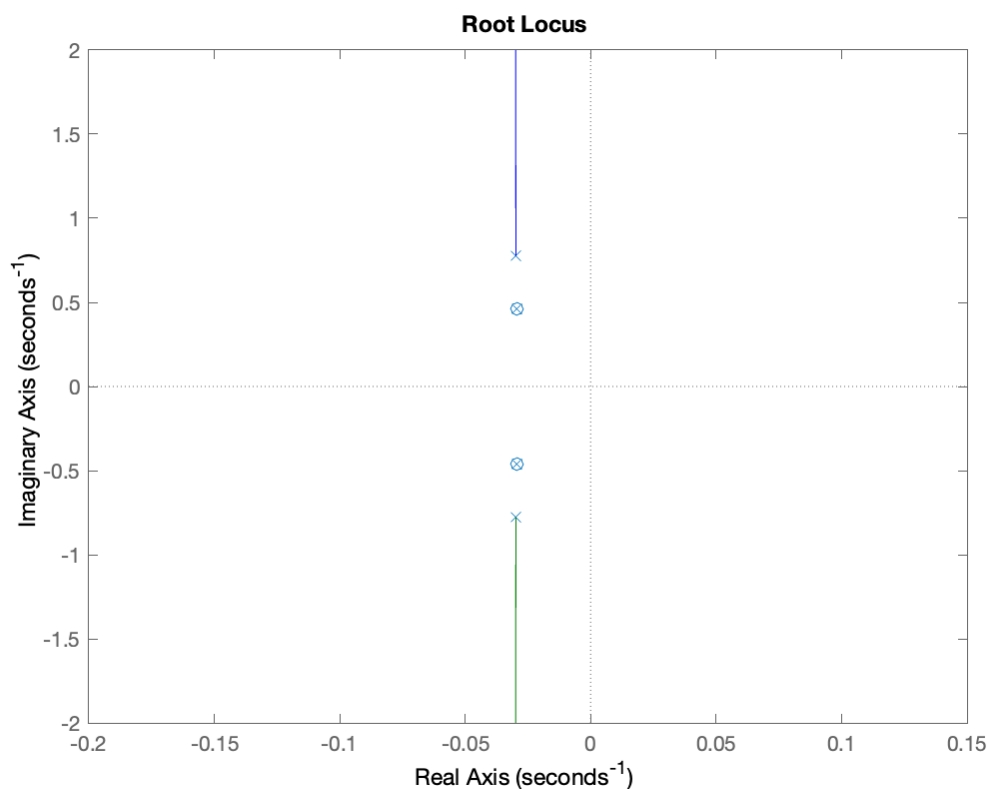


```

sys2 = ss(A2,B2,C,D);
g2 = tf(sys2);
g2
figure(1)
rlocus(g2);

```

$$g2 = \frac{38.84 s^2 + 2.286 s + 8.231}{s^4 + 0.1186 s^3 + 0.8193 s^2 + 0.04828 s + 0.1282}$$



سوال 6

همانطور که میدانیم، برای محاسبه ماتریس انتقال حالت داریم:

$$\varphi(t) = (SI - A)^{-1}$$

و اما برای محاسبه ماتریس انتقال حالت در متلب داریم:

```
phi_s2 = inv(s*eye(4)-A2);
phi_s2 = simplify(phi_s2);
```

محاسبه پاسخ

و همچنین برای

حالت و پاسخ خروجی سیستم، روابط زیر را داریم:

$$x(s) = \varphi(s)x(0) + \varphi(s)Bu(s)$$

$$y(s) = C\varphi(s)x(0) + C\varphi(s)Bu(s) + Du(s)$$

$$U = [1/s \ 1/s \ 1/s \ 1/s];$$

$$X0 = [0; 0.5; 0; 1];$$

$$XS = \text{phi_s1} * X0 + \text{phi_s1} * B * U;$$

$$YS = C * \text{phi_s1} * X0 + C * \text{phi_s1} * B * U + D * U;$$

$$XT = \text{ilaplace}(XS);$$

$$XT = \text{vpa}(\text{simplify}(XT))$$

$$YT = \text{ilaplace}(YS);$$

$$YT = \text{vpa}(\text{simplify}(YT))$$

سوال (7)

چه زمانی مود های سیستم تحریک نمیشود:

همانطور که میدانیم برای اینکه هیچ مودی تحریک نشود اینورس v در $x0$ ضرب شود و جایی که صفر است مود تحریک شده است:

$$\text{inv}(V1) * [0; 0; 0; 1] =$$

$$0.9989$$

$$-0.0252$$

$$0.0011$$

$$-0.0001$$

در اینجا میبینیم به ازای چنین چیزی صفر نمیشود پس با حل معادله و برابری آن با صفر حل میکنیم:

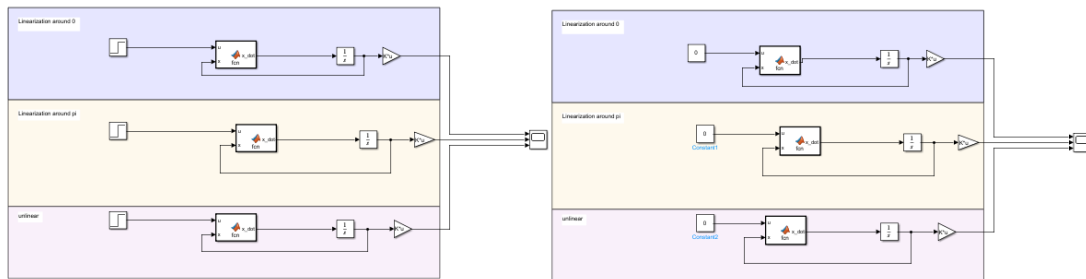
$$\text{eqns_freq} = [\text{inv}(V1) * [x_01; x_02; x_03; x_04] == [0 \ 0 \ 0 \ 0]];$$

$$\text{freq} = \text{solve}(\text{eqns_freq}, [x_01 \ x_02 \ x_03 \ x_04]);$$

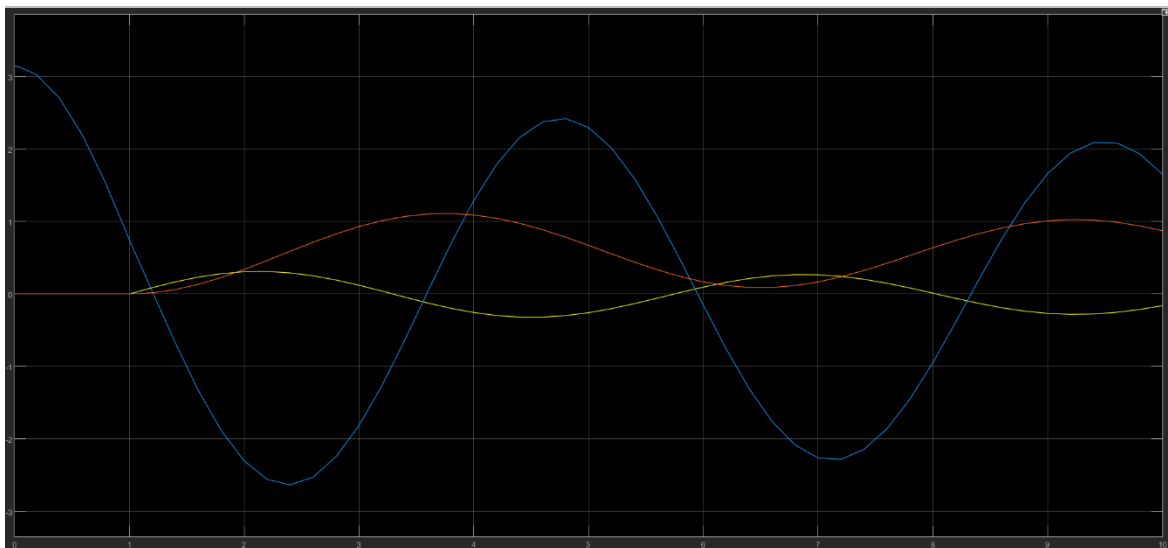
مشاهده میکنیم که مقادیر برابر $[0 \ 0 \ 0 \ 0]$ است.

سوال (8)

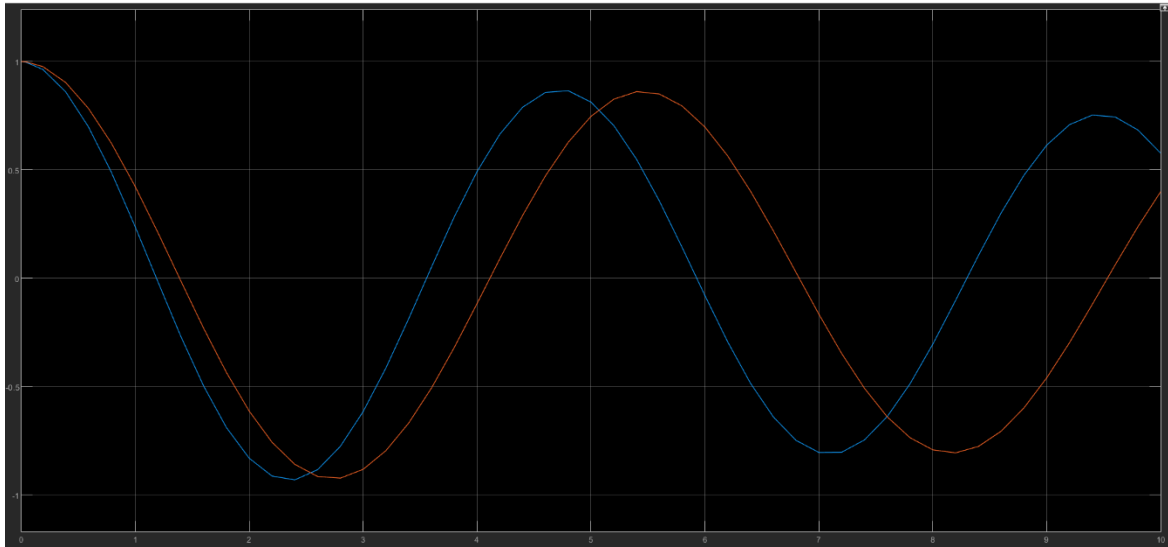
شبیه سازی سیمولینک نیز قرار داده شده است:



خروجی سیمولینک به ازای پاسخ حالت صفر:



خروجی سیمولینک به ازای پاسخ حالت صفر:



سوال 10

تعریف کنترل پذیری (controllability):

سیستم $\dot{x} = Ax + Bu$ را کنترل پذیر گویند اگر سیگنال کنترل $u(t)$ وجود داشته باشد که بتوان حالت سیستم را از هر حالت اولیه $x(t_0) = x_0$ در زمان اولیه t_0 به هر حالت نهایت نهایی $x(t_1) = x_1$ در زمان محدود $t_1 - t_0$ انتقال داد. در این حالت گوییم زوج (A, B) کنترل پذیر است.

ماتریس کنترل پذیری:

$$C(A, B) = [B \quad AB \quad A^2B \quad A^3B \quad \dots \quad A^{n-1}B]$$

شرط لازم و کافی برای کنترل پذیری آن است که ماتریس کنترل پذیری دارای رتبه سطری کامل باشد.

در متلب برای این دستور داریم:

$$C_AB = \text{ctrb}(A, B)$$

$$C_AB =$$

$$\begin{bmatrix} 0 & -16.0100 & 0.9637 & 3.5304 \end{bmatrix}$$

-16.0100	0.9637	3.5304	-0.4353
0	38.8400	-2.3196	-23.3173
38.8400	-2.3196	-23.3137	2.7905

$$\text{rank}(C_{AB}) = 4$$

ماتریس کنترل پذیری ما فول رنک سطری می باشد پس کنترل پذیر می باشد.

رویت پذیری (Observability):

سیستم را رویت پذیر گویند اگر برای هر t_0 زمانی چون $t_1 < \infty$ وجود داشته باشد که:

$$x(t; t_0, x_0, u) = y(t; t_0, \dot{x}_0, u)$$

برابر دو شرط اولیه x_0 و \dot{x}_0 را نتیجه می دهد. اگر این خاصیت به ازای x_0 و t_0 ای برقرار باشد، سیستم را کاملاً رویت پذیر می نامند. در این حالت گوییم زوج (A, C) رویت پذیر است.

ماتریس رویت پذیری:

$$\begin{matrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{matrix}$$

شرط لازم و کافی برای کنترل پذیری آن است که ماتریس کنترل پذیری دارای رتبه ستونی کامل باشد.

در متلب برای این دستور نیز داریم:

$$O_{AC} = \text{obsv}(A, C)$$

$$O_{AC} =$$

0	0	1.0000	0
0	0	0	1.0000
0.0017	0.0005	-0.6032	-0.0595
-0.0002	0.0016	0.0359	-0.5997

$$\text{rank}(C_{AB}) = 4$$

ماتریس رویت پذیری ما فول رنک ستونی می باشد پس رویت پذیر می باشد.

مشاهده میکنیم که رنک ماتریس کنترل پذیری و رویت پذیری کامل است در نتیجه سیستم مینیمال است.

سوال 11

تجزیه کالمن در واقع برای پیدا کردن زیر سیستم های سیستم مورد نظر ما می باشد. از آنجایی که سیستم مورد تحلیل ما دارای مرتبه کامل رویت پذیری و رویت پذیری می باشد، سیستم بطور کامل مینیمال می باشد و زیر سیستمی نخواهیم داشت. پس در نهایت تجزیه کالمن نخواهیم داشت.

فاز دوم پروژه

دستیابی به اهداف پایداری و ردگیری ورودی مرجع

سوال 1)

حالت $\theta = 0$ و $\theta = \pi$:

برای بررسی پایداری لیپانوف میتوانیم ابتدا Q را به صورت ماتریس $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ مینویسیم که مثبت معین باشد سپس میتوانیم با استفاده از دستور `lyap` میتوانیم معادله مقابل را حل کنیم $A^T P + P A = Q$ با حل معادله فوق ماتریس P را به دست می آوریم حال اگر P مثبت معین باشد، سیستم پایدار است.

```
Q = eye(4)
lypa_A1 = lyap(A1,Q);
d_lypa_A1 = eig(lypa_A1);
isposdef_A1 = all(d_lypa_A1 > 0)

lypa_A2 = lyap(A2,Q);
d_lypa_A2 = eig(lypa_A2);
isposdef_A2 = all(d_lypa_A1 > 0);
```

```
isposdef_A1 =

    logical

     1

isposdef_A2 =

    logical

     1
```

برای بررسی با مقادیر ویژه میتوان از دستور `eig()` مقادیر ویژه ها را به دست آورد.

```
eig_A1 =

-0.0295 + 1.3250i
-0.0295 - 1.3250i
-0.0297 + 1.1746i
-0.0297 - 1.1746i

eig_A2 =

-0.0295 + 0.4601i
-0.0295 - 0.4601i
-0.0298 + 0.7761i
-0.0298 - 0.7761i
```

مشاهده میشود که مقادیر ویژه سیستم در سمت چپ ROS میباشند و در این صورت سیستم پایدار است. همچنین اگر سیستم پایدار داخلی باشد، میتوان نتیجه گرفت که سیستم پایدار $BIBO$ نیز می باشد.

سوال (2)

حالت $\theta = 0$ و $\theta = \pi$:

برای بدست آوردن زیر سیستم کاهش ناپذیر خطی، ابتدا باید بررسی کنیم که در تابع تبدیل ما حذف صفر و قطب وجود دارد یا خیر.

برای این کار، رنک ماتریس کنترل پذیری و رویت پذیری را محاسبه می کنیم:

```
>> rank(ctrb(A1,B1))
```

```
ans =
```

4

```
>> rank(observ(A1,C))
```

```
ans =
```

```
4
```

با توجه به این که رنک هر دو ماتریس برابر 4 شده است و رتبه کامل می باشد، تابع تبدیل حاصله قطعا مینیمال می باشد.

سوال (3)

حالت $\theta = 0$:

ابتدا باید مود های سیستم را مورد بررسی قرار دهیم و برای این کار از دستور `eig()` در متلب استفاده می کنیم.

```
>> eigen_values = eig(A1)
```

```
eigen_values =
```

```
-0.0295 + 1.3250i
```

```
-0.0295 - 1.3250i
```

```
-0.0297 + 1.1746i
```

```
-0.0297 - 1.1746i
```

با بررسی مقادیر ویژه، متوجه می شویم که قطب ها نزدیک 0 می باشند و با کوچکترین نویز، اغتشاش و یا تغییرات اندک، سیستم به ناپایداری می رسد. علاوه بر این موضوع، سیستم سرعت مناسبی ندارد و باید به سرعت مناسب آن برسیم. برای رسیدن به این نتایج دلخواه باید مود های سیستم را از محور موهومی دور کنیم.

```
k1=place(A1,B1,[-3+1.2j,-3-1.2j,-3+1.3j,-3-1.3j]);
Anew1=A1-B1*k1;
sys1_feedback=ss(Anew1,B1,C,D);
gnew1=tf(sys1_feedback);
```

مقادیر بدست آمده جدید را نمایش می دهیم:

```
>> k1
```

```
k1 =
```

```
2.5455    16.2067    2.4181    6.9864
```

```
>> gnew1
```

```
gnew1 =
```

$$\frac{38.84 s^2 + 2.286 s + 67.98}{s^4 + 12 s^3 + 57.13 s^2 + 126.8 s + 111.6}$$

حالت $\theta = \pi$:

همانند قبل مود های سیستم را با `eig()` در متلب بدست می آوریم:

```
>> eigen_values = eig(A2)
```

```
eigen_values =  
  
-0.0295 + 0.4601i  
-0.0295 - 0.4601i  
-0.0298 + 0.7761i  
-0.0298 - 0.7761i
```

همانند حالت $\theta = 0$ می بینیم که قطب ها نزدیک 0 می باشند، و تمامی مشکلات حالت θ را برای این حالت نیز داریم. پس باید قطب ها را از محور موهومی دور کنیم :

```
k2=place(A2,B2,[-3+0.8j,-3-0.8j,-3+0.45j,-3-0.45j]);  
Anew2=A2-B2*k2;  
sys2_feedback=ss(Anew2,B2,C,D);  
gnew2=tf(sys2_feedback);
```

مقادیر بدست آمده جدید را نمایش می دهیم :

```
k2 =
```

```
-12.7061   -17.6703   -3.8611   -6.9779
```

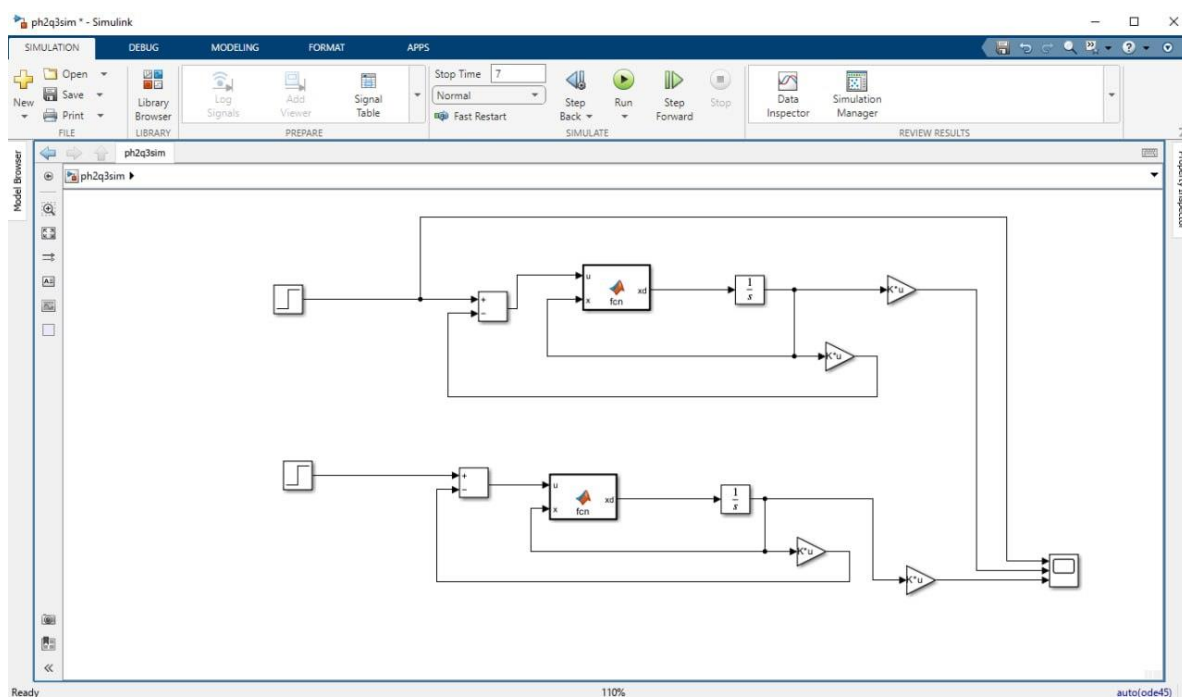
```
>> gnew2
```

```
gnew2 =
```

```
38.84 s^2 + 2.286 s + 8.231  
-----  
s^4 + 12 s^3 + 54.84 s^2 + 113.1 s + 88.71
```

سوال 4)

در سیمولینک بلوک ها را شبیه سازی میکنیم و یک بار تابع تبدیل خطی و بار دیگر تابع غیرخطی را قرار می دهیم. سپس پاسخ پله را مقایسه میکنیم :



کد بلوک خطی :

```

1 function xd= fcn(u,x)
2
3 A1 =[0 1.0000 0 0;
4 -1.7560 -0.0591 0.0109 0.0005;
5 0 0 0 1.0000;
6 0.0139 0.0005 -1.3810 -0.0595];
7
8 B1=[0;
9 -16.0100;
10 0;
11 38.8400];
12
13 xd = A1*x+B1*u;
14

```

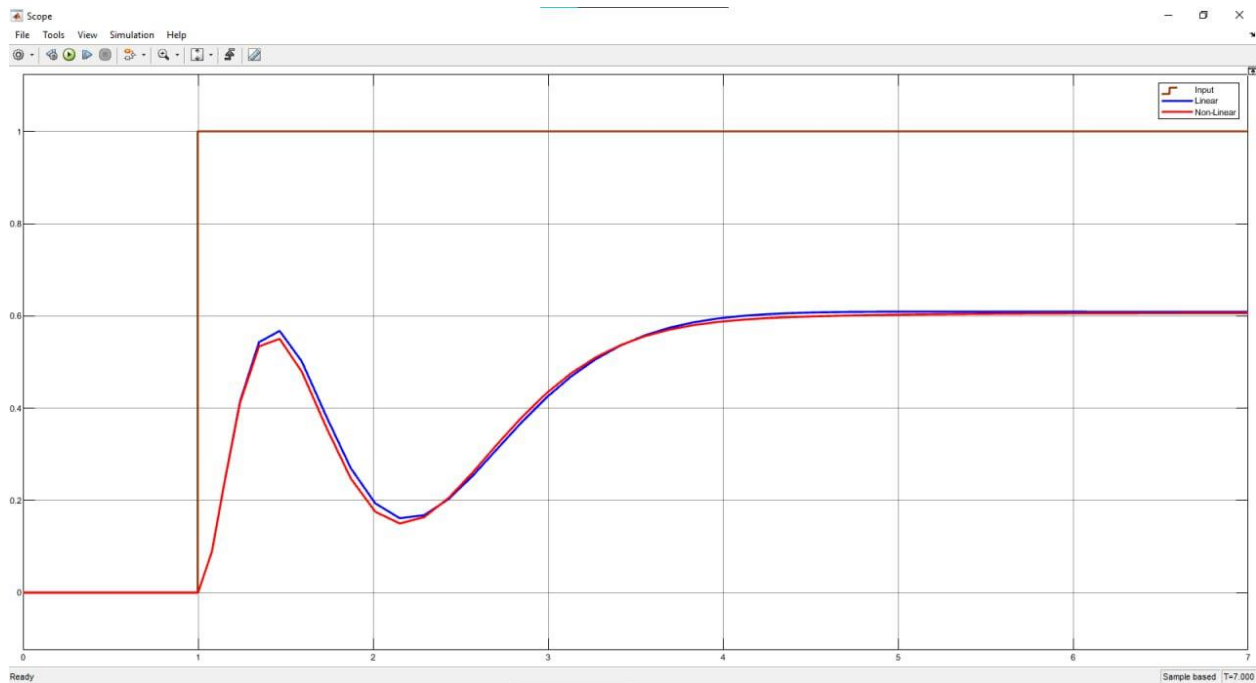
کد بلوک غیر خطی:

```

1 function xd= fcn(u,x)
2   theta1=x(1);
3   theta1_d=x(2);
4   theta2=x(3);
5   theta2_d=x(4);
6
7
8   theta1_dd=(2*cos(theta1 - theta2)*sin(theta1 - theta2)*theta1_d^2 + 1890*theta1_d + 252*sin(theta1 - theta2)*theta2_d^2 ...
9     - 15*cos(theta1 - theta2)*theta2_d + 31500*theta1 + 24696*sin(theta1) - 98*cos(theta1 - theta2)*sin(theta2) ...
10    - 250*theta2*cos(theta1 - theta2))/(2*(cos(theta1 - theta2)^2 - 16002)) - (1601*u)/100;
11   theta2_dd=(971*u)/25 - (254*sin(theta1 - theta2)*theta1_d^2 + 15*cos(theta1 - theta2)*theta1_d ...
12    + 2*cos(theta1 - theta2)*sin(theta1 - theta2)*theta2_d^2 - 1905*theta2_d - 31750*theta2 - 12446*sin(theta2) ...
13    + 196*cos(theta1 - theta2)*sin(theta1) + 250*theta1*cos(theta1 - theta2))/(2*(cos(theta1 - theta2)^2 - 16002));
14
15   xd=zeros(4,1);
16   xd(1)=theta1_d;
17   xd(2)=theta1_dd;
18   xd(3)=theta2_d;
19   xd(4)=theta2_dd;
20

```

خروجی:



در پاسخ پله خروجی مشاهده میکنیم که پاسخ خطی و غیرخطی کاملاً برهم منطبق هستند و نتیجه مطلوبی گرفتیم در نتیجه مدل خطی به خوبی میتواند نماینده مدل غیر خطی در تغییر سیستم به وسیله فیدبک حالت است.

سوال (5)

از انجایی که پاسخ پله سیستم ناپایدار نشد نیازی به پاسخ دادن به سوال ۵ نیست.

سوال (6)

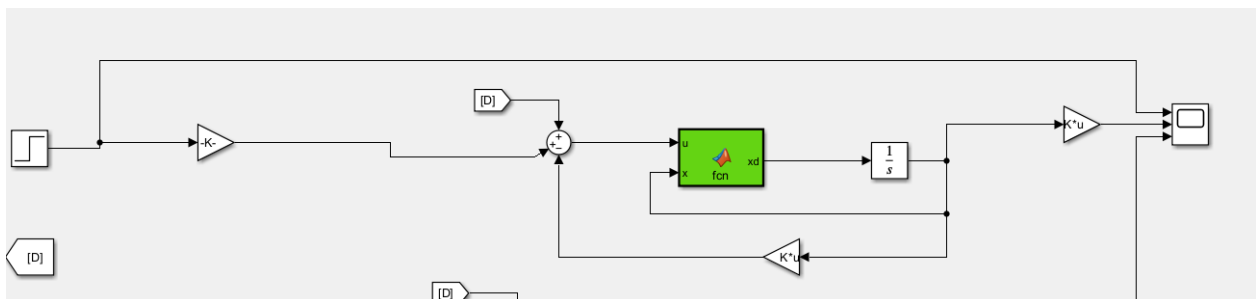
در بحث طراحی پیش جبرانساز استاتیکی کفایت گین مورد نیاز را بدست آوریم. به این منظور تابع تبدیل سیستم با حضور فیدبک حالت را می نویسیم.

حالت $\theta=0$:

```
fx >> gnew1= (38.84 s^2 + 2.286 s +67.98)/(s^4 + 12 s^3 + 57.13 s^2 + 126.8 s + 111.6
```

$$G = gnew(0)^{-1} \rightarrow G = 1.6417$$

حالا باید این گین را به سیستم اعمال کنیم:



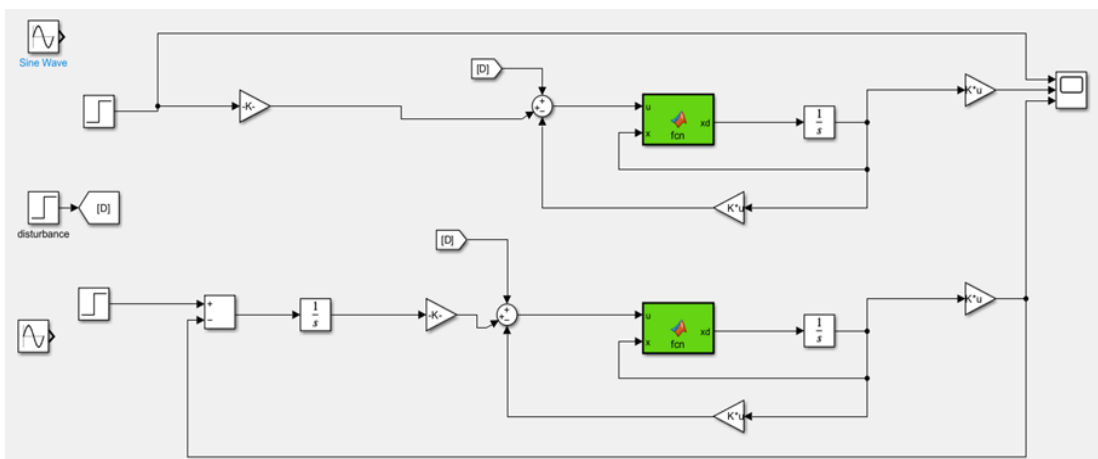
برای طراحی پیش جبران ساز دینامیکی در ابتدا کنترل پذیری را بررسی میکنیم سپس ماتریس های افزوده را تشکیل میدهیم و سپس برای آن ها فیدبک حالت طراحی می کنیم.

```
>> r1=rank([B1 A1;0 C]);

Augmented_A1=[A1 zeros(4,1);-C 0];
Augmented_B1=[B1;0];
Augmented_C=[C 0];

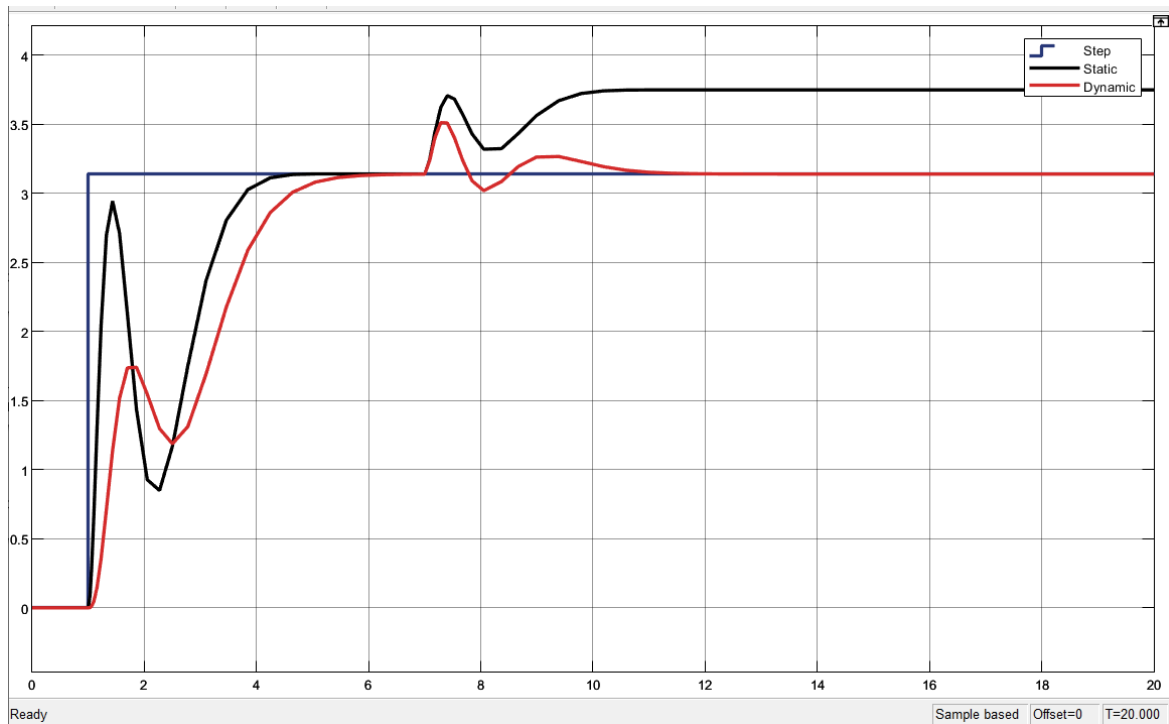
kd1=place(Augmented_A1,Augmented_B1,[-3+1.2j,-3-1.2j,-3+1.3j,-3-1.3j,-2]);
Anewd1=Augmented_A1-Augmented_B1*kd1;
sysld_feedback=ss(Anewd1,Augmented_B1,Augmented_C,D);
gnewd1=tf(sysld_feedback);
```

سیستم با دو پیش جبران ساز استاتیکی و دینامیکی:

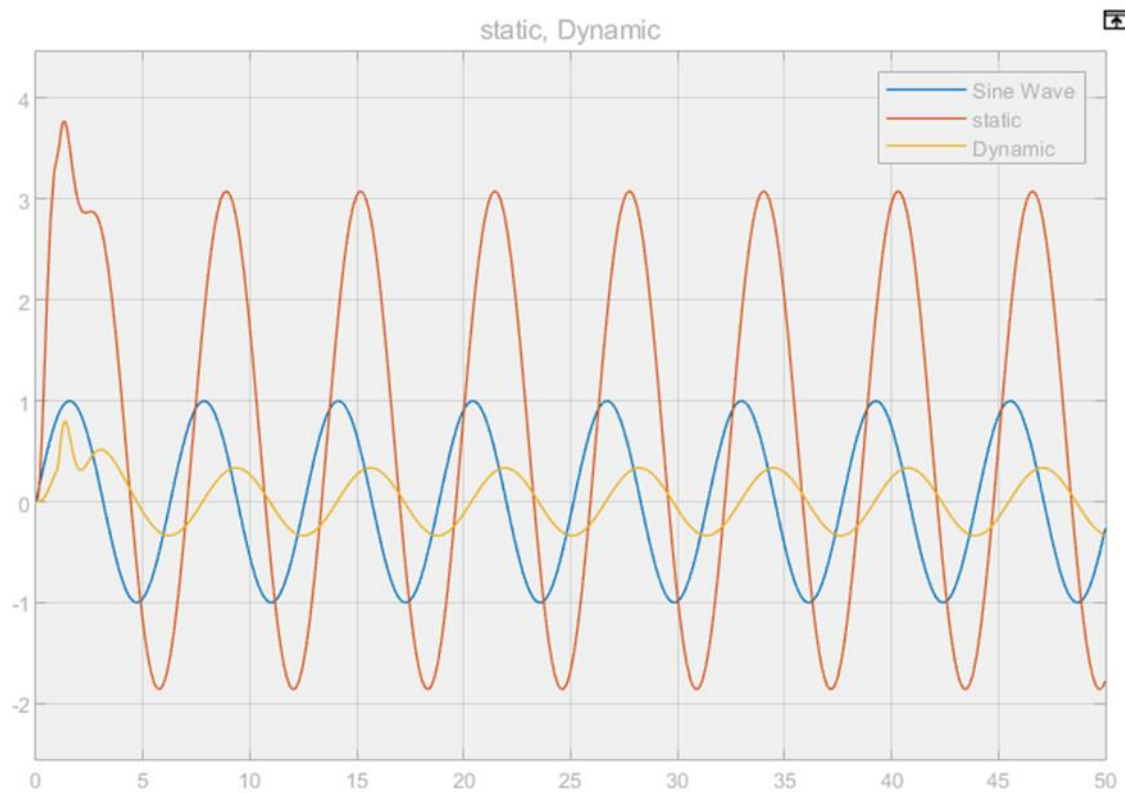


همان طور که ملاحظه می گردد سیستم با پیش جبران ساز استاتیکی پس از اعمال اغتشاش در ثانیه 7 از لحاظ گین نمی تواند ورودی را تعقیب کند ولی پیش جبران ساز دینامیکی، این امکان را هم به ما می دهد.

نتایج scope:



پاسخ به ورودی سینوسی برای سیستم خطی شده حول 0:

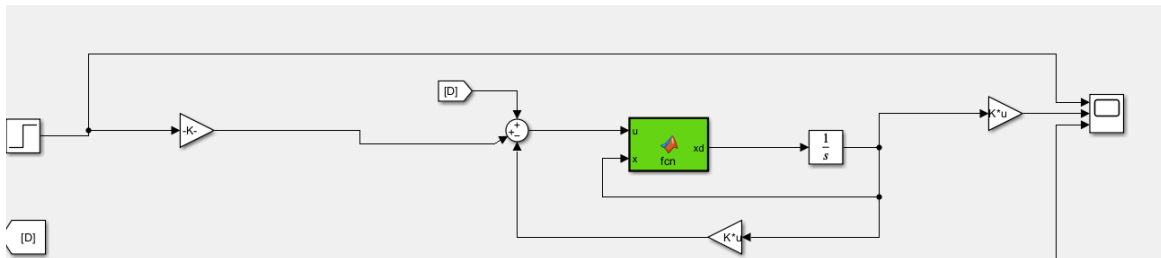


حالت $\theta = \pi$:

```
fx >> gnew2 = (38.84 s^2 + 2.286 s + 8.231)/(s^4 + 12 s^3 + 54.84 s^2 + 113.1 s + 88.71)
```

$$G = gnew(0)^{-1} \rightarrow G = 10.777$$

حالا باید این گین را به سیستم اعمال کنیم:

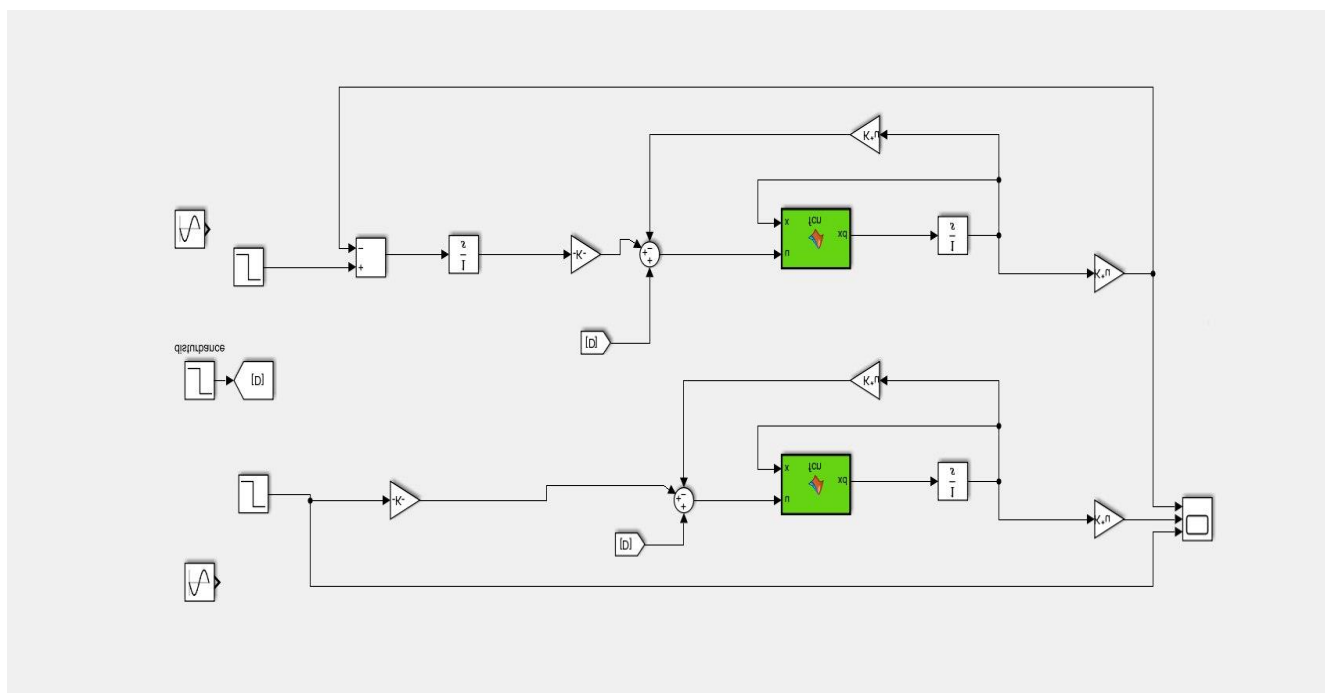


برای طراحی پیش جبران ساز دینامیکی، پس از بررسی کنترل پذیری، ماتریس های افزوده را تشکیل می دهیم و سپس برای آن ها فیدبک حالت طراحی می کنیم.

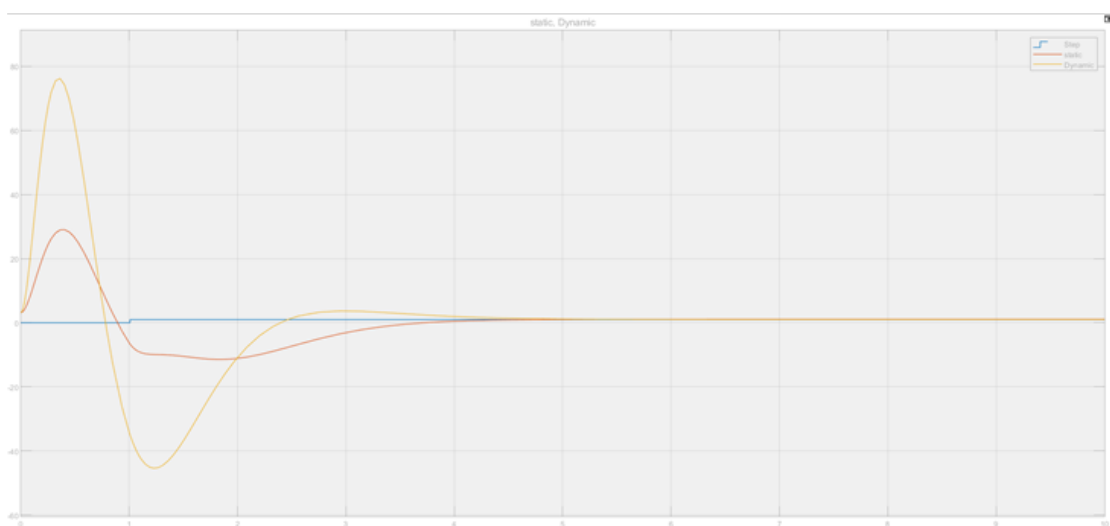
```
fx >>
r1=rank([B2 A2;0 C]);

Augmented_A2=[A2 zeros(4,1);-C 0];
Augmented_B2=[B2;0];
Augmented_C=[C 0];

kd2=place(Augmented_A2,Augmented_B2,[-3+0.8j,-3-0.8j,-3+0.45j,-3-0.45j,-2]);
Anewd2=Augmented_A2-Augmented_B2*kd2;
sys2d_feedback=ss(Anewd2,Augmented_B2,Augmented_C,D);
gnewd2=tf(sys2d_feedback);
```

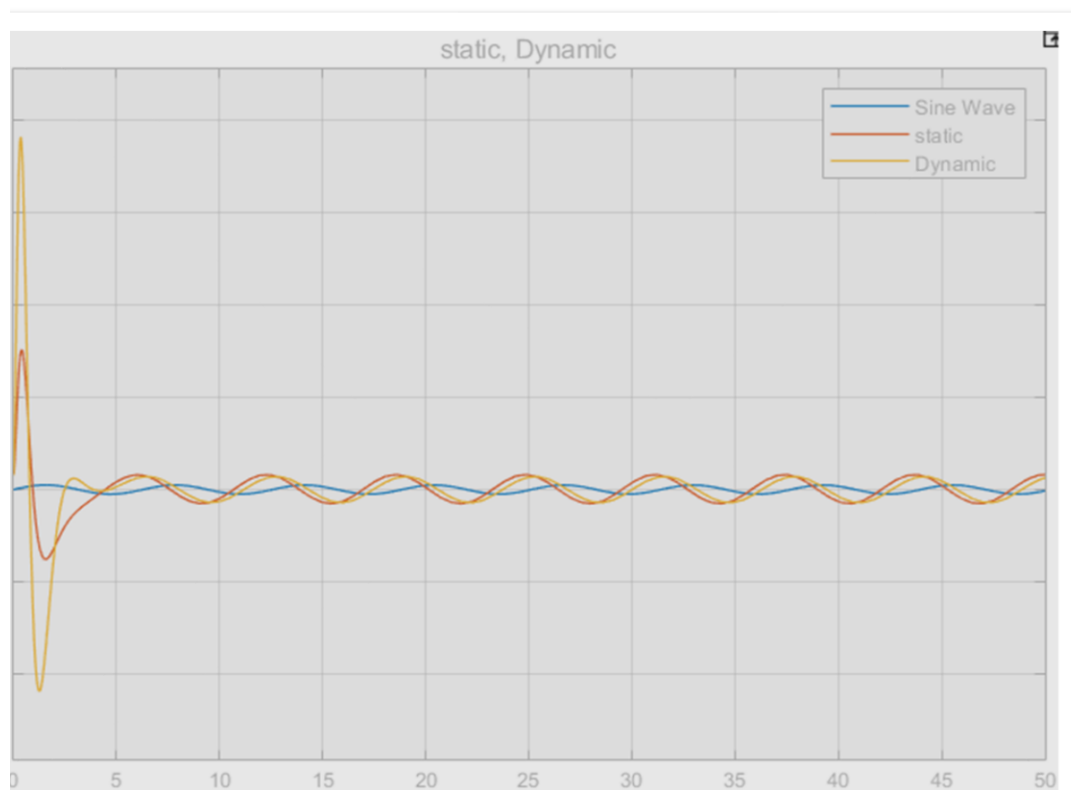


در شکل بالا، کل سیستم را با دو پیش جبران ساز استاتیکی و دینامیکی مشاهده می کنید.



جبران ساز استاتیک اندکی با مقدار اصلی فاصله دارد اما جبران ساز دینامیک آن خطا را نیز صفر کرده است. و اثر اغتشاش در جبران ساز استاتیک کاملاً رفع شده است.

پاسخ برای ورودی سینوسی برای خطی سازی شده حول π :



- [1] Yang, Yuan, Yongjiang Huang, Haoran Yang, Tingting Zhang, Zixuan Wang, and Xixiang Liu. 2021. "Real-Time Terrain-Following of an Autonomous Quadrotor by Multi-Sensor Fusion and Control" *Applied Sciences* 11, no. 3
- [2] Sabatino, Francesco Di. "Quadrotor control: modeling, nonlinear control design, and simulation." (2015).
- [3] نقاش، ابوالقاسم و رحیم زاده، احسان، 1395، کنترل کوادروتور با رگولاتورهای LQR و PID، اولین همایش ملی نگرشی نوین در مهندسی برق و کامپیوتر، کرمانشاه
- [4] Q. Quan, *Introduction to Multicopter Design and Control*, Springer Singapore, 2017.
- [5] Büyüksarıkulak, M. S., "Autopilot design for a quadrotor /," M.S. – Master of Science, Middle East Technical University, 2014.
- [6] Suicmez, E. C., "TRAJECTORY TRACKING OF A QUADROTOR UNMANNED AERIAL VEHICLE (UAV) VIA ATTITUDE AND POSITION CONTROL"., 2014, 10.13140/2.1.4660.0641.
- [7] S. H. Lee, S. H. Kang and Y. Kim, "Trajectory tracking control of Quadrotor UAV", *Proceedings of the ICCAS International Conference on Control, Automation and Systems*, Gyeonggi-do, Korea, October 2011, pp. 281-28
- [8] L. Derafa, T. Madani and A. Benallegue, "Dynamic modelling and experimental identification of four rotors helicopter parameters ", *Proceedings of the IEEE International Conference on Industrial Technology*, Mumbai, India, December 2006, pp. 1834-1839.
- [9] M. Bangura and R. Mahony, "Nonlinear dynamic modeling for high performance control of a quadrotor ", *Proceedings of Australasian Conference on Robotics and Automation*, Wellington, New Zealand, December 2012
- [10] T. Madani and A. Benallegue, "Backstepping control for a quadrotor helicopter", *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, October 2006, pp. 3255-3260
- [11] Guardeno, R., Manuel J. L., and Víctor M. S. 2019. "MIMO PID Controller Tuning Method for Quadrotor Based on LQR/LQG Theory" *Robotics* 8, no. 2: 36. <https://doi.org/10.3390/robotics8020036>

- M. McCrink and J. W. Gregory, "Blade element momentum modeling of low-Re small UAS electric propulsion systems," in 33rd AIAA Applied Aerodynamics Conference, 2015. [12]
- ع. خاکی صدیق, اصول کنترل مدرن, تهران: انتشارات دانشگاه تهران, 1399. [13]
- Arthur Earl Bryson. *Control of Spacecraft and Aircraft*. Princeton university press Princeton, New Jersey, 1993 [14]

پیوست

کد