

به نام خدا



دانشگاه تهران
پردیس دانشکده‌های فنی
دانشکده برق و کامپیوتر



سیستم های کنترل پیشرفته

فاز دوم پروژه

نام و نام خانوادگی : محمد مهدی رحیمی – محمدرضا بختیاری

شماره دانشجویی : 810197510 – 810197468

دی 1400

فهرست سوالات

- بخش اول - طراحی کنترلر برای سیستم خطی شده 3
- 1- 3
- 2- 7
- 3- 9
- 4- 13
- بخش دوم - کالیبراسیون 16
- 1) تبدیل زاویه به مقدار استاندارد: 16
- 2) تبدیل موقعیت به مقدار استاندارد 18
- بخش سوم - سیستم غیرخطی 19
- ملاحظات : 19
- خواسته ها: 20
- 1- 20
- 2- 21
- 3- 24
- 4- 25

بخش اول – طراحی کنترلر برای سیستم خطی شده

-1

ابتدا با استفاده از قطعه کد زیر مکان هندسی صفرها و قطب های تابع تبدیل را به دست آورده و سپس از روی تابع تبدیل به دست آمده ، معادله مشخصه فعلی سیستم را نمایش می دهیم :

```
A = [0 1 0 0;0 -2.2626 -0.3963 0.0082;0 0 0 1;0 3.5649 16.0655 -0.3324] ;
```

```
B = [0;0.2263;0;-0.3565] ;
```

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

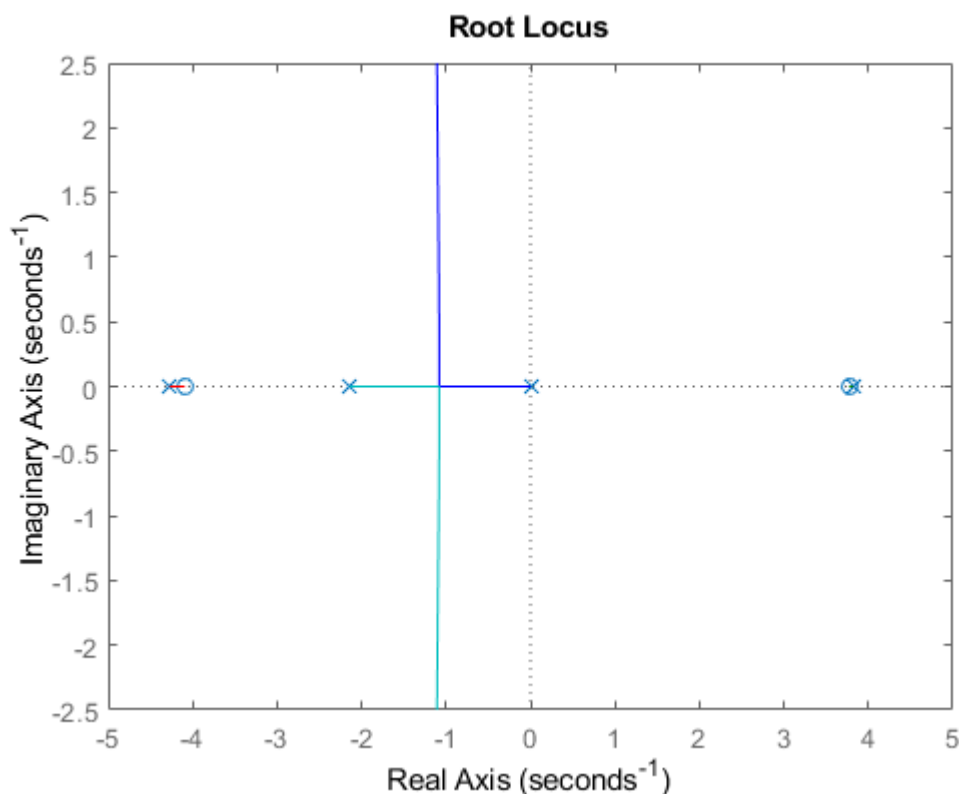
```
D = 0 ;
```

```
[b,a] = ss2tf(A,B,C,D) ;
```

```
G = tf(b,a) ;
```

```
rlocus(G)
```

مکان هندسی به دست آمده به صورت زیر خواهد بود :



شکل 1: مکان هندسی سیستم خطی شده بدون فیدبک حالت

همچنین تابع تبدیل فعلی سیستم به صورت زیر خواهد بود :

$$G(s) = \frac{(0.2263s^2) + (0.0723s) - (3.494)}{s^4 + 2.595s^3 - 15.34s^2 - 34.94s}$$

که در این صورت معادله مشخصه فعلی سیستم به این صورت خواهد بود :

$$\alpha(s) = s^4 + 2.595s^3 - 15.34s^2 - 34.94s$$

که همان طور که مشاهده می کنیم ، تابع تبدیل سیستم دارای 4 قطب بوده که یکی از آن ها ناپایدار است و به طور کلی همانطور که از معادله مشخصه و مکان هندسی پیدا است ، موقعیت قطب های سیستم به این صورت خواهد بود :

$$p_1 = 3.8187 , p_2 = 0 , p_3 = -2.1416 , p_4 = -4.2721$$

معادله مشخصه مطلوب سیستم را طوری طراحی می کنیم که قطب ناپایدار نداشته باشیم (به عبارتی p_1 را به سمت چپ محور موهومی انتقال می دهیم) و همچنین یک بار قطب ها را سعی می کنیم دور از مبدا انتخاب کنیم تا سرعت سیستم نیز بیشتر شود و یک بار نزدیک به مبدا انتخاب کنیم تا سرعت سیستم کاهش یابد .

دسته اول : قطب های سریع

قطب های مطلوب را به این شکل انتخاب می کنیم :

$$p_1 = -10 , p_2 = -9 , p_3 = -8 , p_4 = -7$$

و معادله مشخصه مطلوب نیز به این صورت خواهد بود :

$$\varphi_1(s) = s^4 + 34s^3 + 431s^2 + 2414s + 5040$$

دسته دوم : قطب های کند

قطب های مطلوب را به این شکل انتخاب می کنیم :

$$p_1 = -2 , p_2 = -1.5 , p_3 = -1 , p_4 = -0.5$$

و معادله مشخصه مطلوب نیز به این صورت خواهد بود :

$$\varphi_2(s) = s^4 + 5s^3 + 8.75s^2 + 6.25s + 1.5$$

با استفاده از روش بس و گیورا ضرایب مناسب k را برای هر حالت پیدا می کنیم :

برای این کار ابتدا ماتریس Ψ را تشکیل می دهیم :

+1.000	+2.595	-15.34	-34.94
+0.000	+1.000	+2.595	-15.34
+0.000	+0.000	+1.000	+2.595
+0.000	+0.000	+0.000	+1.000

سپس Ψ^{-1} برابر خواهد شد با :

+1.000	-2.595	+22.07	-62.14
+0.000	+1.000	-2.595	+22.07
+0.000	+0.000	+1.000	-2.595
+0.000	+0.000	+0.000	+1.000

حال ماتریس کنترل پذیری C را محاسبه می کنیم :

$$C = \begin{bmatrix} 0 & 0.2263 & -0.5149 & 1.314 \\ 0.2263 & -0.5149 & 1.314 & -3.4043 \\ 0 & -0.3565 & 0.9252 & -7.8706 \\ -0.3565 & 0.9252 & -7.8706 & 22.1649 \end{bmatrix}$$

در نتیجه C^{-1} برابر خواهد شد با :

$$C^{-1} = \begin{bmatrix} 9.9975 & 4.5979 & 0.0003 & 0.1136 \\ 4.3909 & -0.6517 & -0.1501 & -0.4137 \\ -0.7426 & -0.3016 & -0.5326 & -0.1914 \\ -0.2862 & -0.0059 & -0.1829 & -0.0038 \end{bmatrix}$$

و اما دستور دیگری در نرم افزار متلب وجود دارد که با دادن ماتریس های A و B و جای مد نظر قطب ها به ما مقادیر K را می دهد. در روش قبلی که توضیح دادیم به روش دستی آکرمین مقادیر بدست می آمد اما با این دستور یعنی دستور Place مقادیر را محاسبه کرده و از آن ها استفاده می کنیم .

با توجه به موارد گفته شده خواهیم داشت :

$$K_1 = (\varphi_1 - \alpha) * \psi^{-1} * C^{-1} = 1000 * [-1.4423 \quad -0.7308 \quad -2.3160 \quad -0.552]$$

$$K_2 = (\varphi_2 - \alpha) * \psi^{-1} * C^{-1} = [-0.4293 \quad -11.7981 \quad -70.2510 \quad -14.2354]$$

اگر از روش بس و گیورا استفاده کنیم قطعه کد زیر را به کار می بریم :

% Find K

Controllability = [B A*B A^2*B A^3*B] ;

Psi = [1 2.595 -15.34 -34.94; 0 1 2.595 -15.34; 0 0 1 2.595; 0 0 0 1] ;

Alfa_1 = [34 431 2414 5040] ;

Alfa_2 = [5 8.75 6.25 1.5] ;

a = [2.595 -15.34 -34.94 0] ;

K1 = (Alfa_1-a)*inv(Psi)*inv(Controllability) ; **% Fast pole**

K2 = (Alfa_2-a)*inv(Psi)*inv(Controllability) ; **% Slow pole**

اگر از دستور Place متلب استفاده کنیم خواهیم داشت :

p1 = [-10 -9 -8 -7];

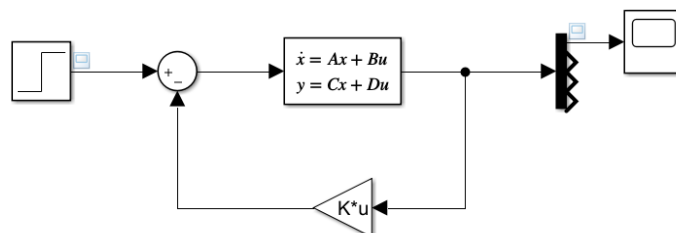
p2 = [-2 -1.5 -1 -0.5];

KK1 = place(A,B,p1);

KK2 = place(A,B,p2);

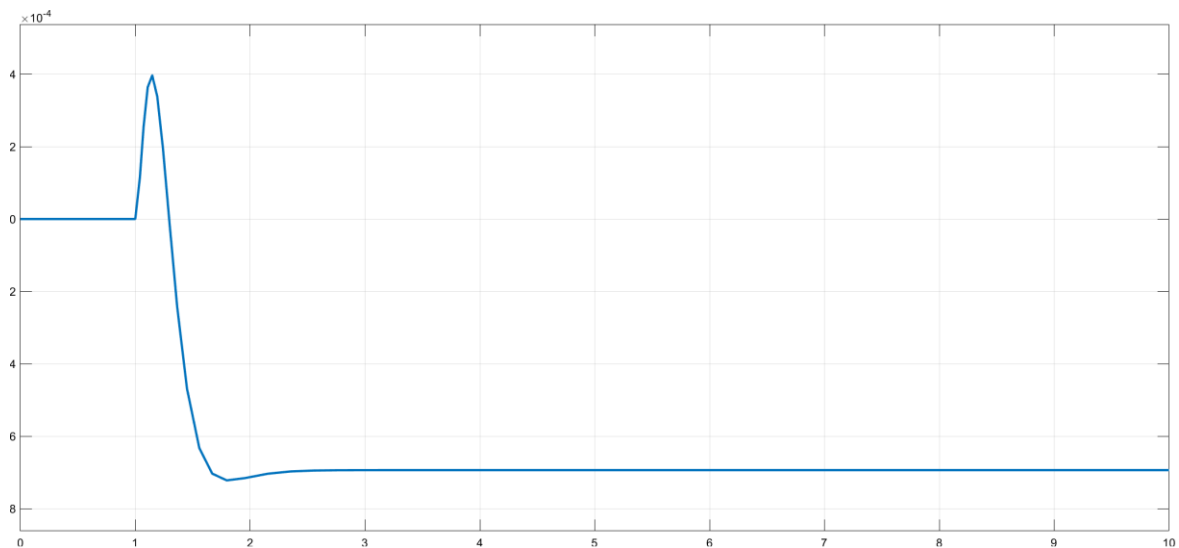
که ضرایب K به دست آمده در هر دو حالت یکسان می باشند .

حال اگر سیستم زیر را برای این بخش پیاده کنیم داریم:



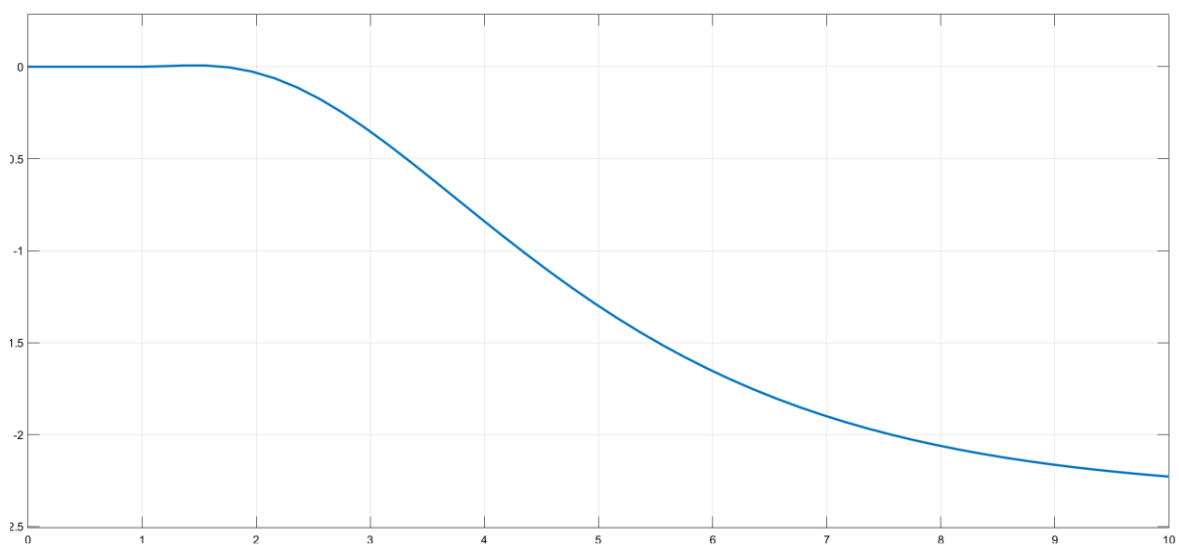
شکل 2: شماتیک کلی سیستم

حال اگر مقادیر K به دست آمده برای حالت تند را قرار دهیم خروجی به شکل زیر می شود :



شکل 3: خروجی حالت تند به ازای ورودی پله

حال اگر K مربوط به حالت کند را قرار دهیم داریم :



شکل 4: خروجی حالت کند به ازای ورودی پله

مشاهده می کنیم در حالتی که قطب های سیستم دور از مبدا قرار گرفته است سیستم با یک بالازدگی¹ قابل ملاحظه ای به سرعت به خروجی پایدار رسیده (تقریباً 2 ثانیه) و در حالتی که قطب ها را نزدیک به مبدا انتخاب کرده ایم ، سرعت سیستم کاهش یافته و چیزی حدود 8 برابر حالت قبل زمان بیشتری نیاز داریم (تقریباً 16 ثانیه) تا خروجی به مقدار پایدار خود برسد .

¹ Overshoot

با توجه به اینکه بخش ردیاب نداریم هیچکدام از حالت ها به مقدار نهایی مناسب نمی رسند و هر کدام به مقدار نهایی متفاوتی می رسند. همانطور که دیده شد در سیستم سریع تر مقادیر بدست آمده برای K بسیار بزرگ بوده و برای پیاده سازی عملی آن نیاز به هزینه بیشتری می باشد.

-2

در این بخش با توجه به اینکه باید کنترل پذیری انتگرالی را بررسی کنیم ابتدا ماتریس های مورد نیاز را با استفاده از ماتریس های اولیه به دست می آوریم :

$$\bar{A} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} \quad \bar{B} = \begin{bmatrix} B \\ 0 \end{bmatrix} \quad \bar{C} = [C \quad 0]$$

حال با توجه به ماتریس های به دست آمده ماتریس کنترل پذیری را محاسبه می کنیم که داریم :

Controllability =

0	0.2263	-0.5149	1.3140	-3.4043
0.2263	-0.5149	1.3140	-3.4043	11.0033
0	-0.3565	0.9252	-7.8706	22.1649
-0.3565	0.9252	-7.8706	22.1649	-145.9492
0	0	-0.2263	0.5149	-1.3140

>> rank(Controllability)

ans =

5

همانطور که مشاهده شد ماتریس کنترل پذیری را تشکیل دادیم و چون ماتریس رتبه کامل می باشد یعنی کنترل پذیر می باشد .

حال مراحل سوال قبل را برای ماتریس های جدید بدست آمده تکرار می کنیم . در این بخش قطب ها را به شکل زیر در نظر می گیریم :

$$p_1 = -11, \quad p_2 = -10, \quad p_3 = -9, \quad p_4 = -8, \quad p_5 = -7$$

که معادله مشخصه آن به شکل زیر می شود:

$$\varphi(s) = s^5 + 45s^4 + 805s^3 + 7155s^2 + 31594s + 55440$$

حال برای محاسبه به ماتریس Ψ و Ψ^{-1} که به ترتیب برابر است با :

1.0000	2.5950	-15.3400	-34.9400	0
0	1.0000	2.5950	-15.3400	-34.9400
0	0	1.0000	2.5950	-15.3400
0	0	0	1.0000	2.5950
0	0	0	0	1.0000
1.0000	-2.5950	22.0740	-62.1494	409.2239
0	1.0000	-2.5950	22.0740	-62.1494
0	0	1.0000	-2.5950	22.0740
0	0	0	1.0000	-2.5950
0	0	0	0	1.0000

و با استفاده از فرمول زیر مقادیر K بدست می آید :

$$K = (\varphi - \alpha) * \psi^{-1} * C^{-1} = 1000 * [-9.37 \quad -3.279 \quad -8.915 \quad -2.201 \quad 15.866]$$

که در اصل داریم :

$$K_1 = 1000 * [-9.369 \quad -3.279 \quad -8.915 \quad -2.201] , \quad K_2 = 15866$$

با استفاده از قطعه کد زیر یک بار با روش بس و گیورا و یک بار با اسفاده از دستور Place ضرایب مورد

نظر را پیدا می کنیم :

روش بس و گیورا :

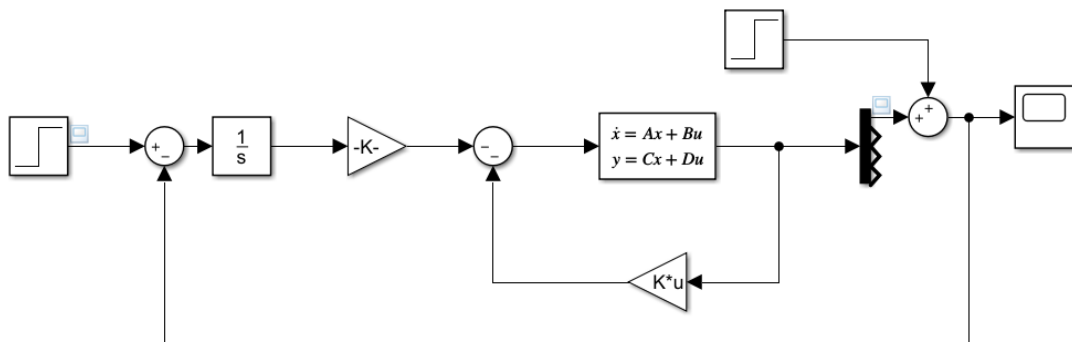
```
A = [0 1 0 0;0 -2.2626 -0.3963 0.0082;0 0 0 1;0 3.5649 16.0655 -0.3324] ;
B = [0;0.2263;0;-0.3565] ;
C = [1 0 0 0] ;
D = 0 ;
z = zeros(4,1);
AA = [A z ; -C 0];
BB = [B;0];
CC = [C 0];
[b,a] = ss2tf(AA,BB,CC,D) ;
G = tf(b,a) ;
Controllability = [BB AA*BB AA^2*BB AA^3*BB AA^4*BB] ;
Psi = [1 2.595 -15.34 -34.94 0;0 1 2.595 -15.34 -34.94;0 0 1 2.595 -15.34;0 0 0 1 2.595;0 0 0 0 1] ;
Alfa = [45 805 7155 31594 55440];
aa = a(2:6);
K = (Alfa-aa)*inv(Psi)*inv(Controllability) ;
K1 = K(1:4);
K2 = K(5) ;
```

با استفاده از دستور Place :

```
poles = [-11 -10 -9 -8 -7];
KK = place(AA,BB,poles);
```

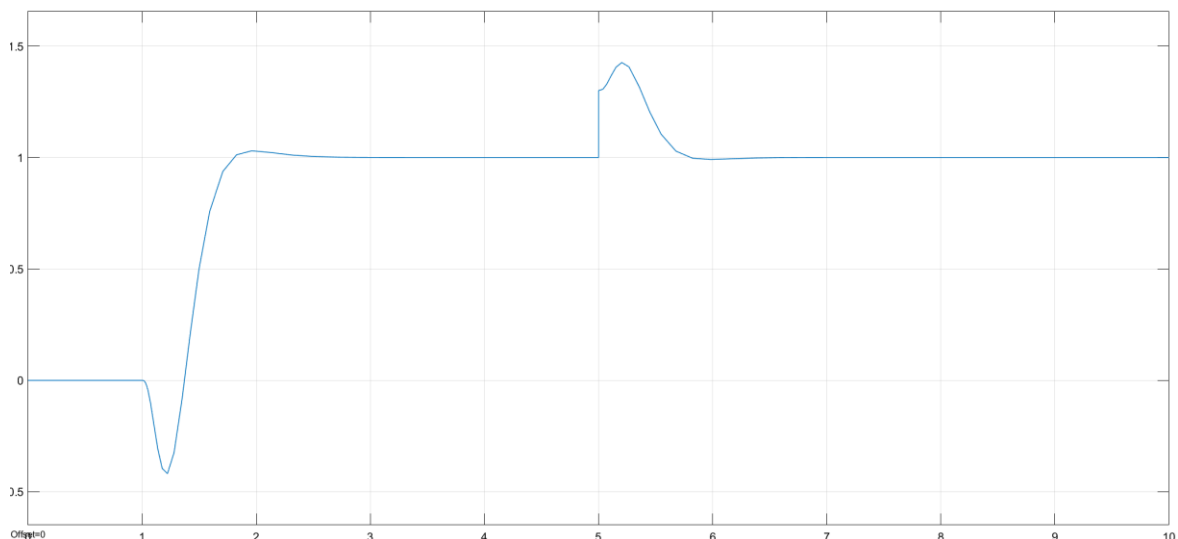
که در هر دو حالت ضرایب K به دست آمده یکسان می باشد .

حال با استفاده از مقادیری که برای K بدست آمد سیستم زیر را طراحی کرده و داریم :



شکل 5: شماتیک کلی سیستم کنترلر انتگرالی

در مدار خروجی مرجع پله ای می باشد که در ثانیه اول و با اندازه یک وارد می شود و اغتشاش نیز پله ای به اندازه 0.3 که در ثانیه 5 وارد می شود می باشد و خروجی به شکل زیر می باشد :



شکل 6: خروجی سیستم به ازای ورودی پله و اعمال اغتشاش

همانطور که مشاهده می شود سیستم نسبت به اغتشاش مقاوم می باشد و عملکرد سیستم بسیار مطلوب می باشد .

-3

در این بخش برای بدست آوردن تخمینگر ابتدا باید ماتریس L را محاسبه کنیم که برای این کار ابتدا مرحله زیر را داریم :

$$A \rightarrow A^T \quad B \rightarrow C^T \quad C \rightarrow B^T$$

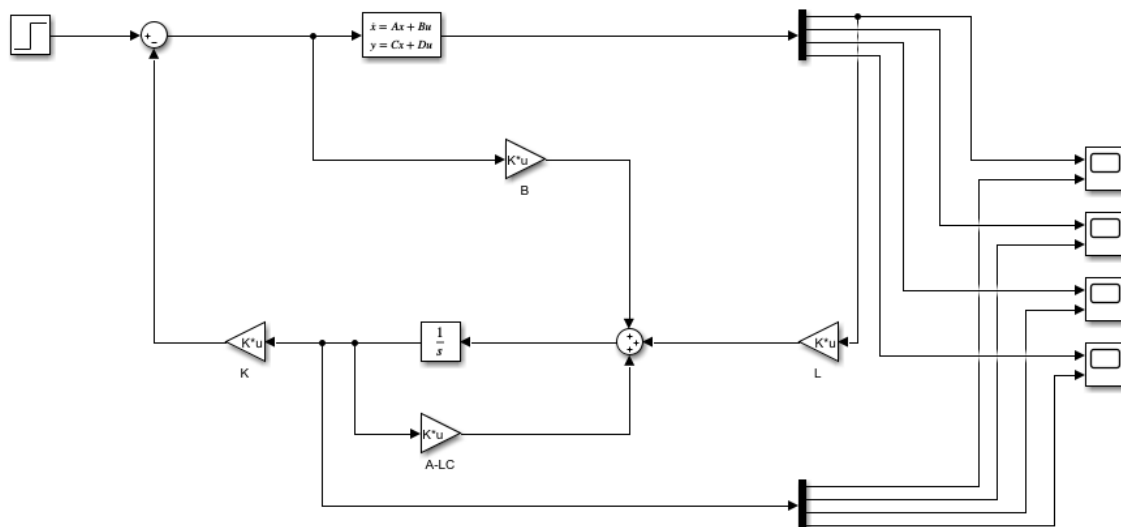
حال با توجه به ماتریس های جدید اگر مراحل سوال قبل را تکرار کنیم به راحتی ماتریس L به دست می آید :

$$L = [31.4 \quad 364.8 \quad -7716 \quad -30280]$$

که با کد زیر محاسبه می شود :

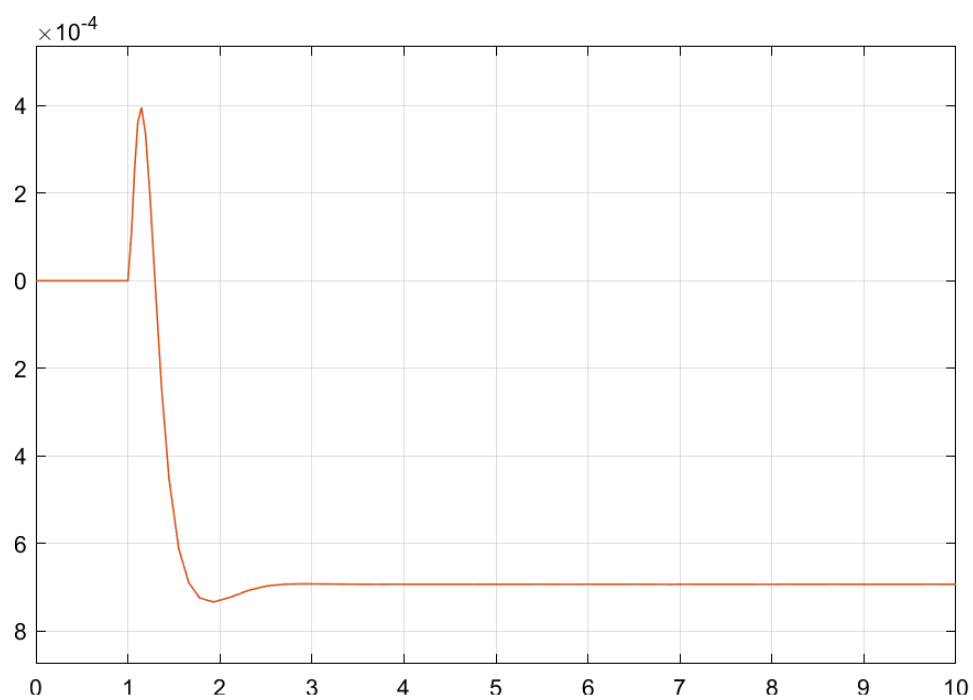
```
A = [0 1 0 0; 0 -2.2626 -0.3963 0.0082; 0 0 0 1; 0 3.5649 16.0655 -0.3324];
B = [0; 0.2263; 0; -0.3565];
C = [1 0 0 0];
D = 0;
z = zeros(4,1);
AA = A';
BB = C';
CC = B';
[b,a] = ss2tf(AA,BB,CC,D);
G = tf(b,a);
Controllability = [BB AA*BB AA^2*BB AA^3*BB];
Psi = [1 2.595 -15.34 -34.94; 0 1 2.595 -15.34; 0 0 1 2.595; 0 0 0 1];
Alfa = [34 431 2414 5040];
aa = a(2:5);
K = (Alfa-aa)*inv(Psi)*inv(Controllability);
L = K';
%%
poles = [-10 -9 -8 -7];
LL = place(AA,BB,poles);
```

حال با توجه به مقدار L بدست آمده و با توجه به اصل جدایی پذیری سیستم ها از همان ماتریس K که در سوال یک بدست آوردیم استفاده می کنیم و داریم :

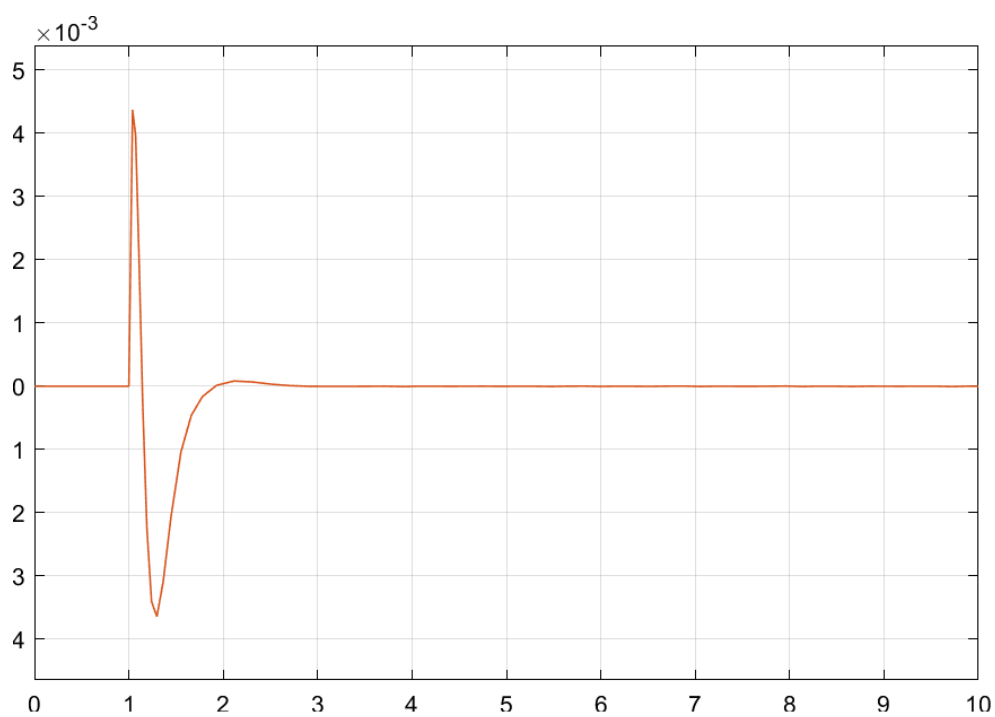


شکل 7: شماتیک کلی سیستم

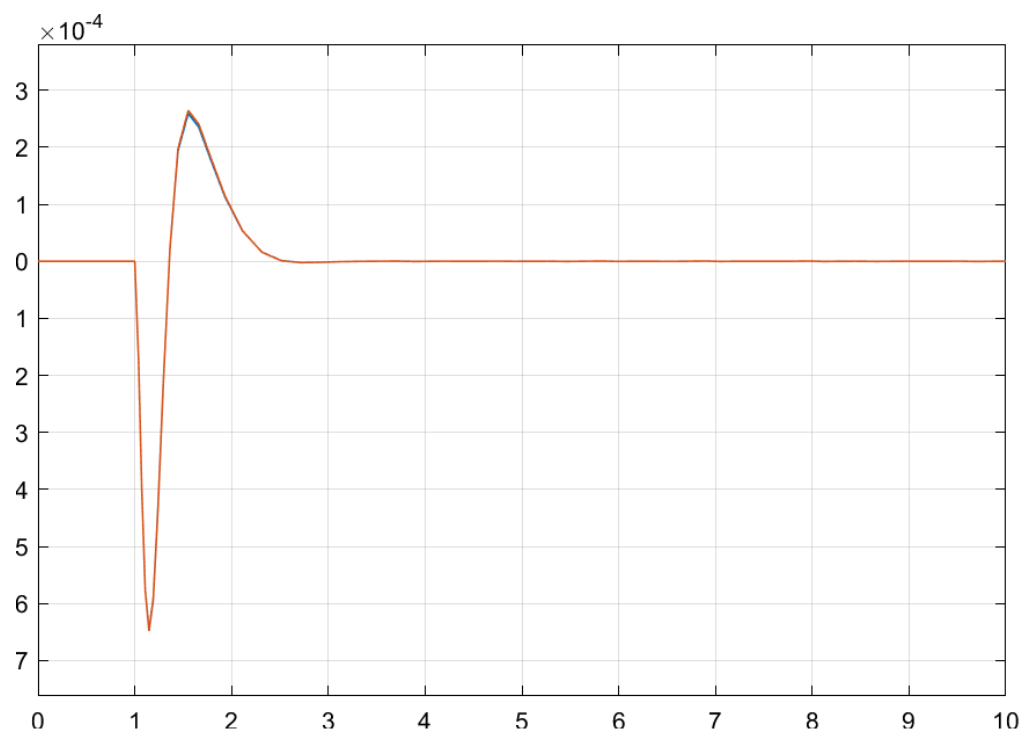
که در این سیستم بخش تخمینگر از ماتریس L استفاده شده و در بخش دیگر از ماتریس K که در سوال یک بدست آوردیم استفاده می کنیم. و مود های سیستم را به ترتیب داریم: (نمودار آبی تخمینگر و نمودار نارنجی سیگنال سیستم اصلی)



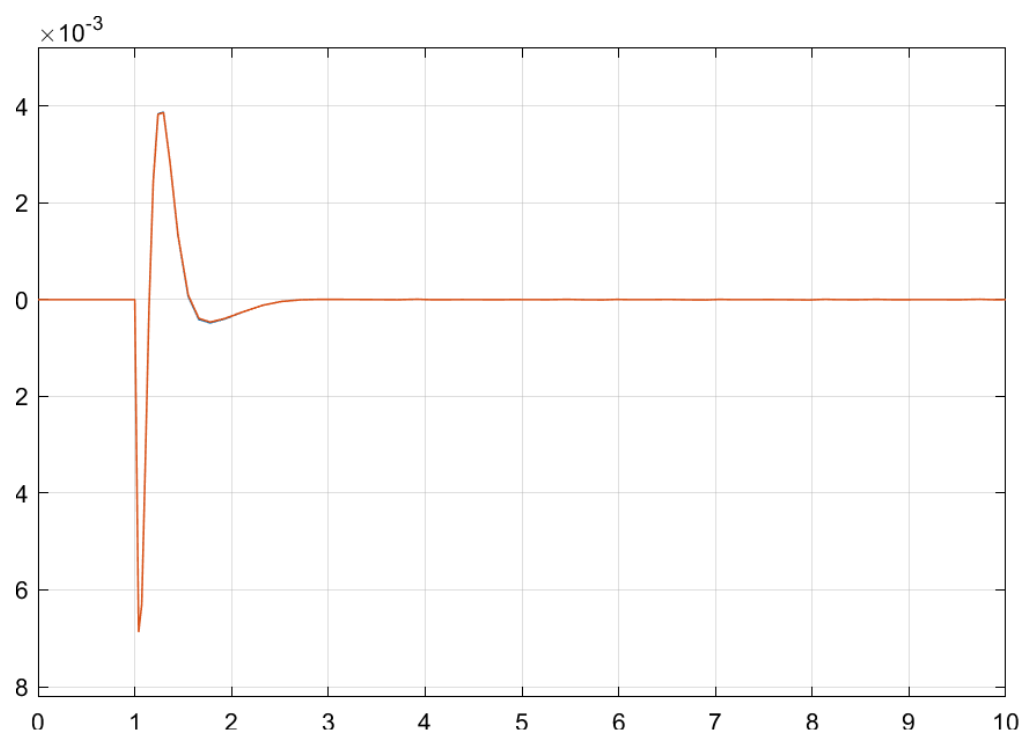
شکل 8: نمودار تخمین و مقدار مود اول سیستم



شکل 9: نمودار تخمین و مقدار مود دوم سیستم



شکل 10: نمودار تخمین و مقدار مود سوم سیستم



شکل 11: نمودار تخمین و مقدار مود چهارم سیستم

همانطور که مشاهده می شود به خوبی تخمین زده شده است و نمودار ها بر روی یکدیگر قرار گرفته اند.

-4

به کمک معادله لیاپانوف طراحی را انجام می دهیم .

ابتدا ماتریس پایدار $F_{3 \times 3}$ را طوری انتخاب می کنیم که مقدار ویژه مشترک با A نداشته باشد .

$$F = \begin{bmatrix} -2.66 & 1 & -0.66 \\ -1.33 & -1 & 0.66 \\ -1.083 & 1.25 & -2.33 \end{bmatrix}$$

که مقادیر ویژه آن به ترتیب 3- , 2- , 1- می باشد که همگی منفی بوده و هیچ مقدار ویژه مشترکی با A ندارد .

حال ماتریس $L_{3 \times 1}$ را طوری انتخاب می کنیم که (F,L) کنترل پذیر باشد :

$$L = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

ماتریس کنترل پذیری را تشکیل می دهیم و بررسی می کنیم که (F,L) کنترل پذیر می باشد یا نه :

$$C = \begin{bmatrix} 1 & -2.66 & 9.5 \\ 2 & -1.33 & 1.16 \\ 3 & -5.58 & 14.25 \end{bmatrix} \rightarrow \text{Rank}(C) = 3$$

در نتیجه (F,L) کنترل پذیر می باشند .

حال معادله لیاپانوف زیر را حل کرده و ماتریس $T_{3 \times 4}$ را به دست می آوریم :

$$TA - FT = LC$$

که از حل معادله خواهیم داشت :

$$T =$$

$$\begin{bmatrix} 0.7500 & -5.9394 & 0.4575 & -0.2107 \\ 2.5278 & -3.6648 & 0.3477 & -0.1401 \\ 2.2917 & -1.8954 & 0.2395 & -0.0853 \end{bmatrix}$$

حال بررسی می کنیم که آیا ماتریس P ویژه است یا خیر و در صورتی که ویژه بود مجدداً ماتریس L متفاوتی انتخاب می کنیم :

$$P = \frac{C}{T}$$

$$P =$$

$$\begin{bmatrix} 1.0000 & 0 & 0 & 0 \\ 0.7500 & -5.9394 & 0.4575 & -0.2107 \\ 2.5278 & -3.6648 & 0.3477 & -0.1401 \\ 2.2917 & -1.8954 & 0.2395 & -0.0853 \end{bmatrix}$$

مشاهده می کنیم که ماتریس فوق غیر ویژه بوده (دترمینان مخالف صفر دارد) و می توانیم با استفاده از معادلات زیر تخمین \hat{X} را بدست آوریم :

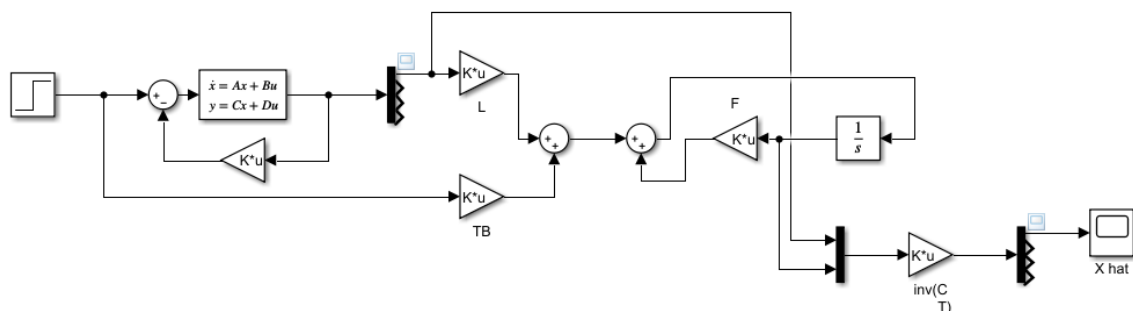
$$\dot{Z} = FZ + TBu + Ly$$

$$\hat{x} = \begin{matrix} C^{-1} & y \\ T & z \end{matrix}$$

ماتریس های مورد نظر را با استفاده از قطعه کد زیر به دست آوردیم :

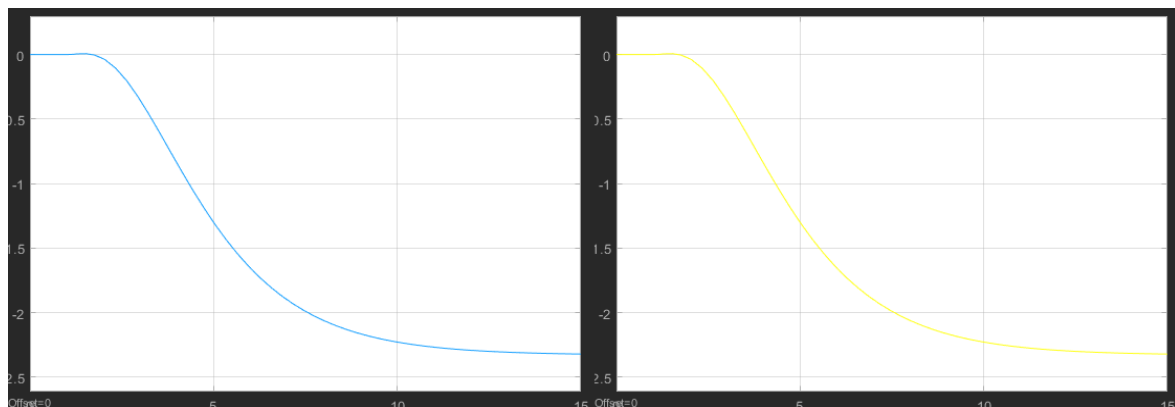
```
A = [0 1 0 0; 0 -2.2626 -0.3963 0.0082; 0 0 0 1; 0 3.5649 16.0655 -0.3324] ;
C = [1 0 0 0] ;
q = [1 2 3; 3 2 1; 2 1 3] ;
F = [-1 0 0; 0 -2 0; 0 0 -3] ;
F = q*F*inv(q) ;
L = [1; 2; 3] ;
Controllability = [L F*L F^2*L] ;
T = lyap(-F,A,-L*C) ;
P = [C;T] ;
```

حال سیستم مورد نظر را پیاده سازی می کنیم :



شکل 12 : شماتیک کلی سیستم

حال با اعمال ورودی پله متغیر حالت اصلی سیستم و متغیر حالت تخمین یافته را نمایش می دهیم :



شکل 13 : به ترتیب از چپ به راست متغیر حالت اصلی و متغیر حالت تخمین یافته

مشاهده می کنیم سیستم طراحی شده به خوبی توانست متغیر حالت مورد نظر (مکان) را تخمین بزند .

برای سایر متغیر های حالت نیز همین روال را انجام می دهیم با این تفاوت که در هر مرحله فقط ماتریس C تغییر می کند و سایر مراحل همانند گذشته است .

از آنجایی که در این بخش ذکر شده است که فقط میخواهیم مکان پاندول معکوس را کنترل کنیم , از نمایش سایر متغیر های حالت خودداری کردیم .

بخش دوم – کالیبراسیون

در این بخش باید دو انکودر را طراحی کنیم که به ترتیب داریم :

1) تبدیل زاویه به مقدار استاندارد:

با استفاده از قطعه کد زیر تبدیل ها و نگاشت های لازم را برای تبدیل زاویه به مقدار استاندارد انجام

می دهیم :

```
function y = fcn(u)
temp = -u / (4096*4) ;
temp = temp*(2*pi) ;
temp = temp - pi ;           % Start point

while 1                      % Set teta between -pi and +pi
    if (temp >= -pi)&&(temp <= pi)
        break
    end
    if temp < -pi
        temp = temp + 2*pi ;
    end
    if temp > pi
        temp = temp - 2*pi ;
    end
end

if (temp >= -pi)&&(temp <= (-pi/2))    % 3 ----> 1
    temp = temp + pi ;
end
if (temp >= (pi/2))&&(temp <= pi)    % 4 ----> 2
    temp = temp - pi ;
end

y = temp ;
```

ابتدا دور روتور را به درجه تبدیل کرده و با ضرب در علامت منفی ، جهت ساعتگرد را مثبت در نظر می گیریم . سپس مقدار فعلی را از زاویه ی مبدا (π) کم می کنیم و در ادامه زاویه را استاندارد سازی می کنیم به این صورت که بین π و $-\pi$ قرار گیرد .

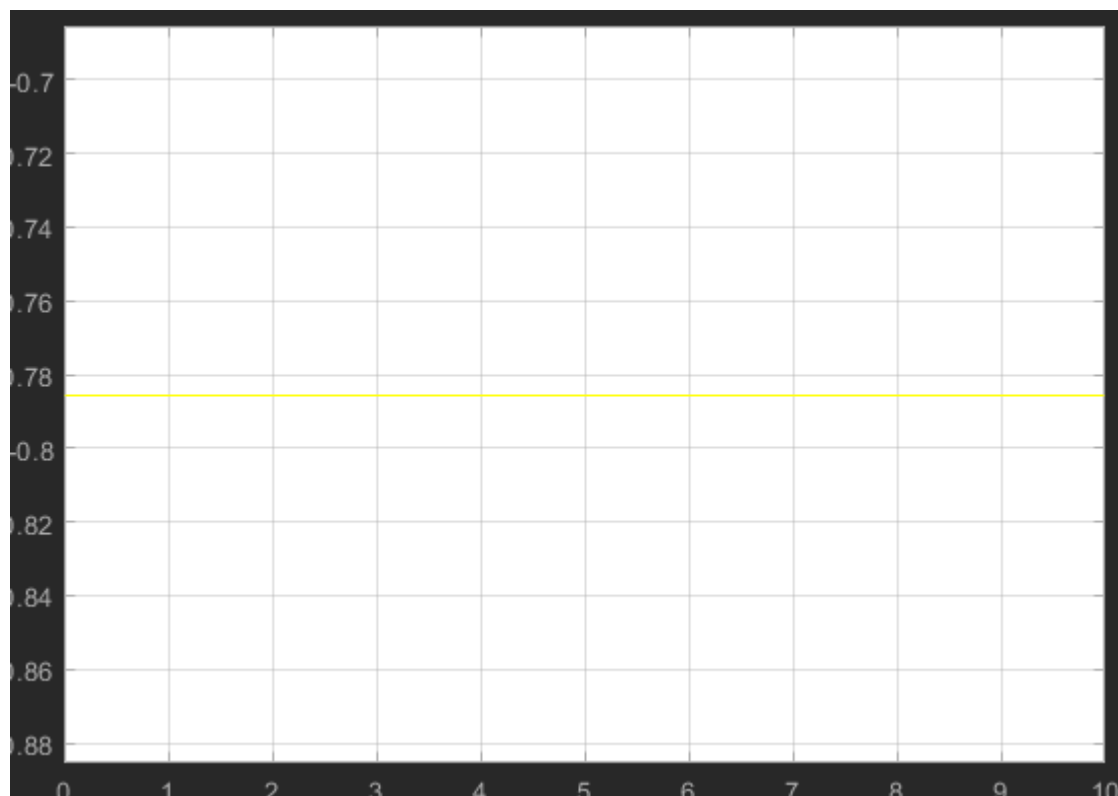
در انتها نگاشت نهایی از ربع 3 و 4 به ترتیب به ربع های 1 و 2 را انجام می دهیم .

با استفاده از شبیه سازی صحت عملکرد تابع را بررسی می کنیم :



شکل 14 : تابع مورد نظر برای تبدیل زاویه به مقدار استاندارد

به عنوان مثال ورودی ثابت $0.5*4096$ را به تابع می دهیم و خروجی را مشاهده می کنیم :



شکل 15 : خروجی تابع مورد نظر به ازای ورودی $0.5*4096$

همان طور که مشاهده کردیم خروجی به دست آمده صحیح می باشد . (خروجی به ازای ورودی $0.5*4096$ برابر با -0.78 شد که از قبل هم همین انتظار را داشتیم)

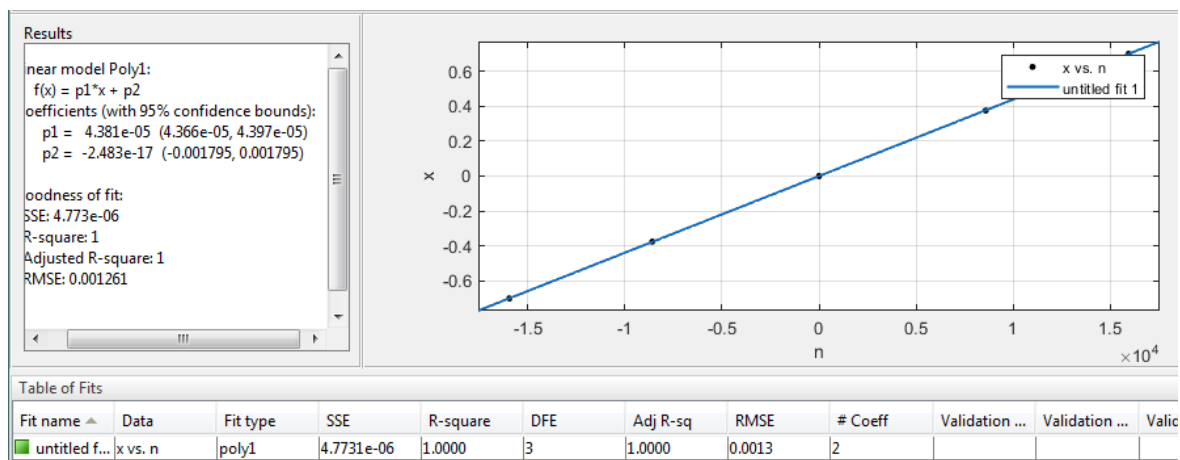
تمام خطی سازی ها و مراحل که تا به اینجا انجام دادیم بر روی سیستم خطی سازی شده بوده است که این خطی سازی را ما حول زاویه صفر می باشد. اگر سیستم ما با نقطه تعادلی که برای آن در نظر گرفتیم و خطی سازی را حول آن انجام داده باشیم فاصله داشته باشد مراحل که با سیستم خطی سازی شده انجام دادیم عملکرد خوبی نخواهد داشت. با توجه به مواردی که توضیح دادیم اگر زاویه سیستم در ربع سوم و چهارم قرار بگیرند با نقطه تعادل فاصله زیادی داشته و سیستم عملکرد خوبی نخواهد داشت. بنابراین این سیستم را به ربع اول و دوم برده تا به نقطه تعادل نزدیک شده و بتوان از مراحل که بر اساس سیستم خطی سازی شده طی کردیم استفاده کرد.

2) تبدیل موقعیت به مقدار استاندارد

در این بخش با استفاده از داده های داده شده و ابزار cftool متلب خطی را برازش می دهیم . این خط قابلیت این را دارد که با دانستن مقدار n یا همان سیگنال ورودی بر حسب دور می تواند میزان جا به جایی را بر حسب متر بیان کند . ابتدا داده ها را وارد متلب کرده سپس از دستور cftool استفاده می کنیم .

```
x = [-0.7 -0.375 0 0.375 0.7];
n = [-15960 -8590 0 8590 15960];
cftool(n,x)
```

اگر کد را اجرا کنیم پنجره مربوط به این ابزار باز شده و خطی را برای ما برازش می دهد که دو پارامتر خط برازش شده به همراه خطای آن را بیان می کند .



شکل 16: خط برازش شده به همراه پارامترهای آن

همانطور که مشاهده می شود خط برازش شده به شکل زیر می باشد :

$$x = 4.381 \times 10^{-5} \times n - 2.483 \times 10^{-17}$$

بنابراین رابطه ای که برای تبدیل خروجی آنکودر موقعیت به مقدار استاندارد بود را بدست آوردیم .

بخش سوم – سیستم غیر خطی

ملاحظات :

1- ابتدا بلوک های کالیبراسیون طراحی شده بخش دو را به صورت زیر بر روی خروجی انکودر اعمال می کنیم :



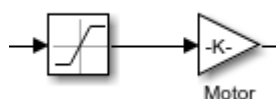
شکل 17 : اعمال بلوک های کالیبراسیون بخش 2 بر روی خروجی انکودر

2- به علت وجود نویز و انواع نایقینی نیاز به طراحی یک فیلتر پایین گذر داریم تا نویز را که در فرکانس های بالا به صورت تقویت شده موجود است ، عبور نکند . این فیلتر پایین گذر را به سادگی و با ایجاد یک قطب دور از مبدا می توانیم ایجاد کنیم ، در اینجا از قطب در 100- استفاده کردیم . فیلتر مذکور را نیز دقیقاً بعد از بلوک های کالیبراسیون قرار می دهیم تا اجازه عبور نویز حاصل از زاویه بر حسب رادیان و مکان بر حسب متر را ندهیم .

در یک سامانه ی بلادرنگ ، استفاده از این فیلتر ها منجر به اضافه شدن یک قطب می شود به این صورت که منجر به اضافه شدن تاخیر اضافی به سیستم می شود و در نتیجه سیگنال فیدبک ارسال شده دارای تاخیر بوده که نهایتاً سیستم با تاخیر می تواند موقعیت خود را تشخیص دهد در نتیجه در واقعیت ممکن است نتوانیم سیستم را به درستی کنترل کنیم .

3- با قرار دادن بلوک Saturation اجازه نمی دهیم تا ولتاژ ورودی بیشتر از حد مجاز به سیستم اعمال شود .

4- در نهایت برای تبدیل ولتاژ اعمالی به نیرو از یک بهره 4.375 استفاده می کنیم .



شکل 18 : اعمال بلوک اشباع و بهره برای تبدیل ولتاژ به نیرو

خواسته ها:

-1

از قبل ماتریس های K را برای قطب تند و کند به دست آورده ایم و کافی است ماتریس های P را برای هر کدام محاسبه کنیم .

برای قطب تند و کند ماتریس K را به ترتیب داریم :

$$K_1 = (\varphi_1 - \alpha) * \psi^{-1} * C^{-1} = 1000 * [-1.4423 \quad -0.7308 \quad -2.3160 \quad -0.552]$$

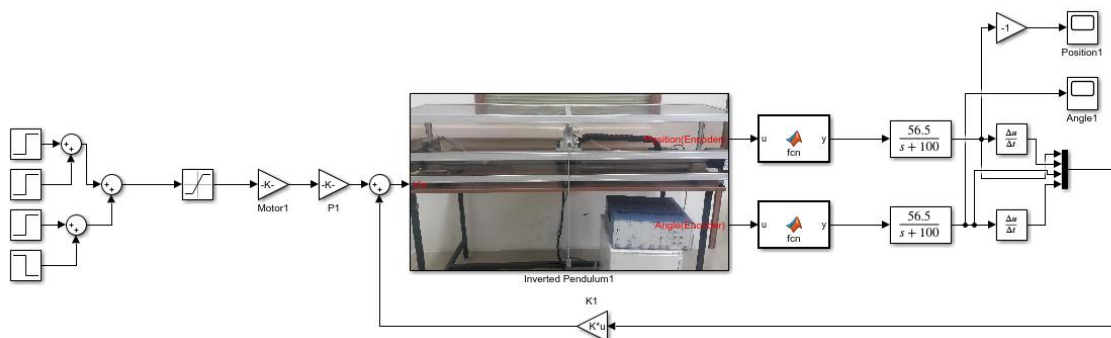
$$K_2 = (\varphi_2 - \alpha) * \psi^{-1} * C^{-1} = [-0.4293 \quad -11.7981 \quad -70.2510 \quad -14.2354]$$

و ماتریس P نیز برای قطب تند و کند نیز به صورت زیر است :

$$P_1 = [-C(A - BK_1)^{-1}B]^{-1} = -1442.3$$

$$P_2 = [-C(A - BK_2)^{-1}B]^{-1} = -0.4293$$

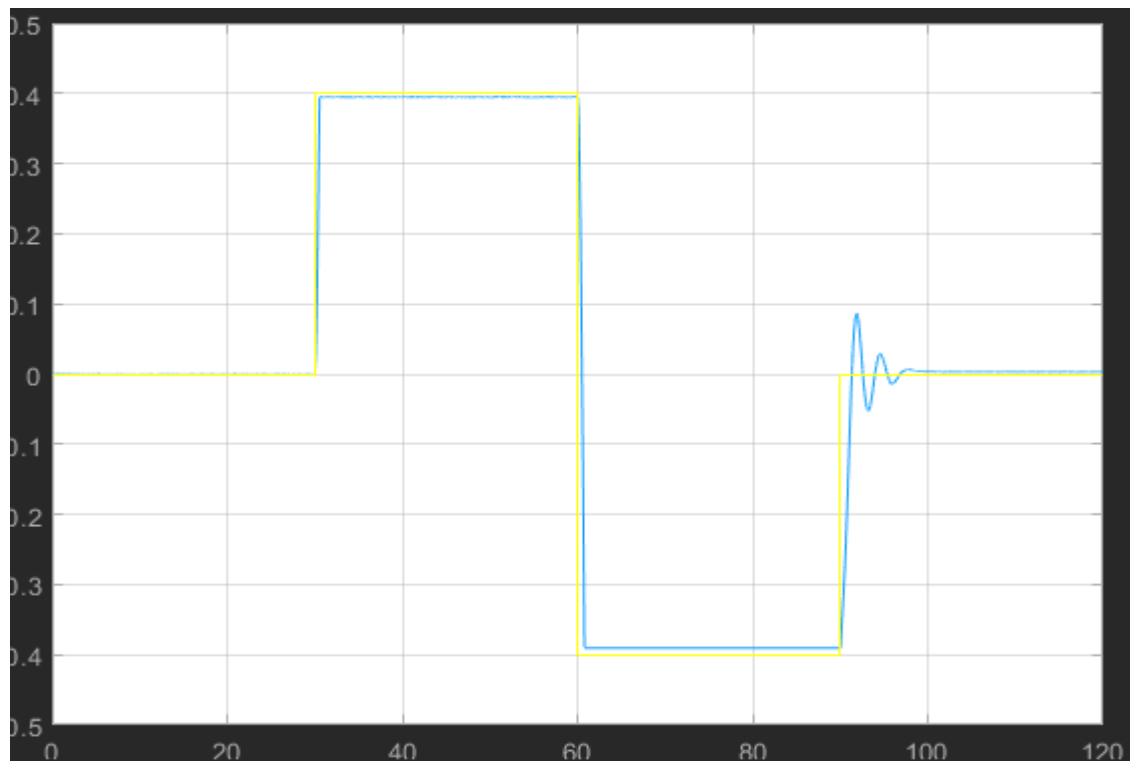
در ادامه سیستم زیر را پیاده سازی می کنیم :



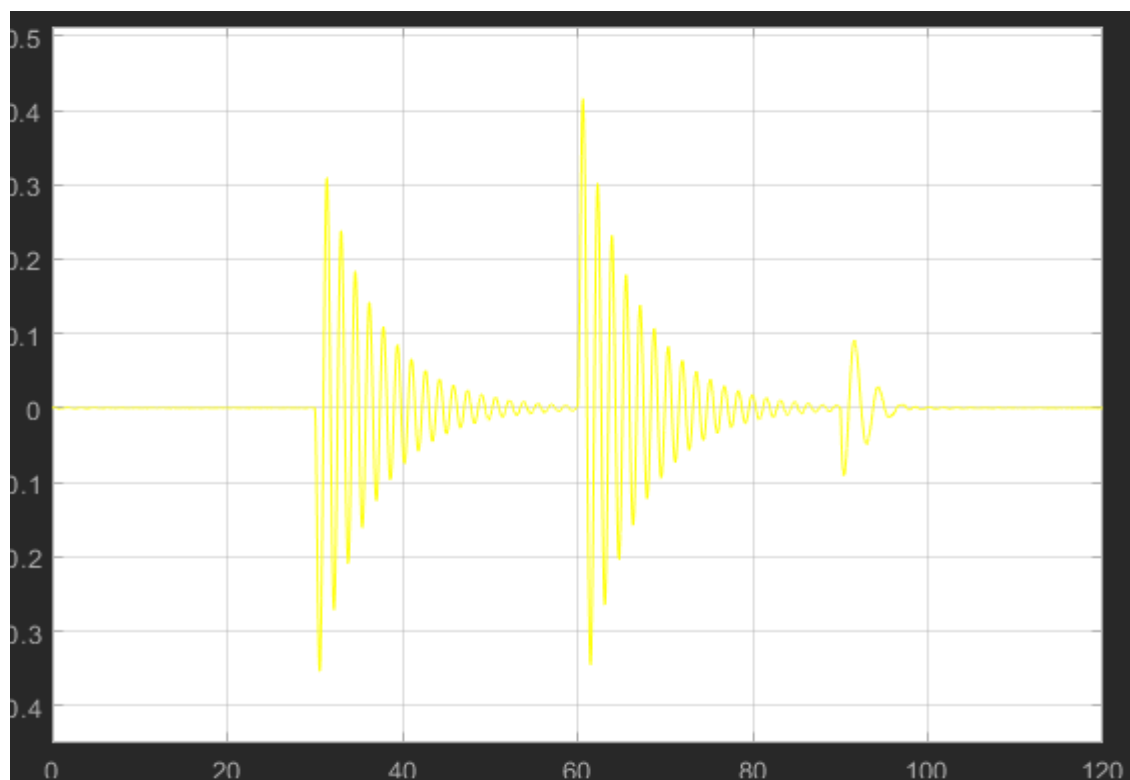
شکل 19: نمای کلی از سیستم غیر خطی به همراه ردیاب

همانگونه که مشاهده می کنیم در سیستم غیر خطی دیگر به حالت های سیستم دسترسی نداریم ،
 فلذا با استفاده از خروجی های سیستم متغیر های حالت را تشکیل می دهیم .

ابتدا قطب های تند را قرار داده و خروجی موقعیت ارابه و زاویه را مشاهده می کنیم :

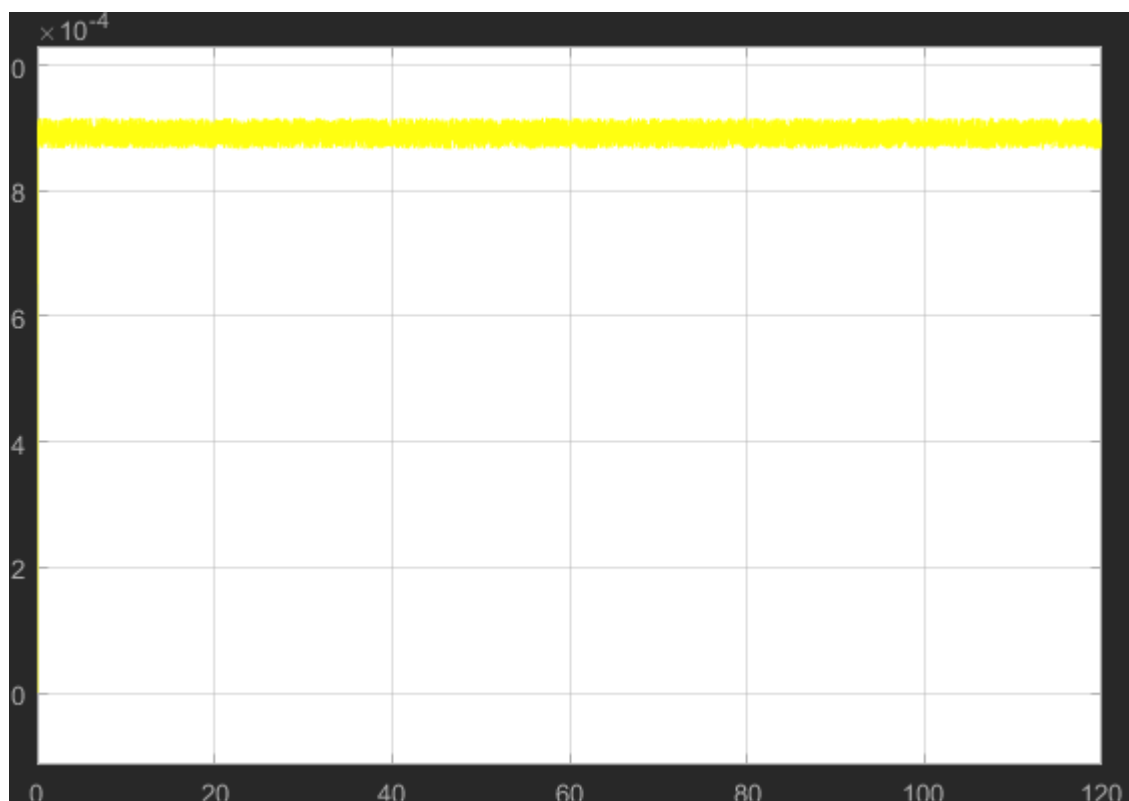


شکل 20 : موقعیت ارابه به ازای قطب تند

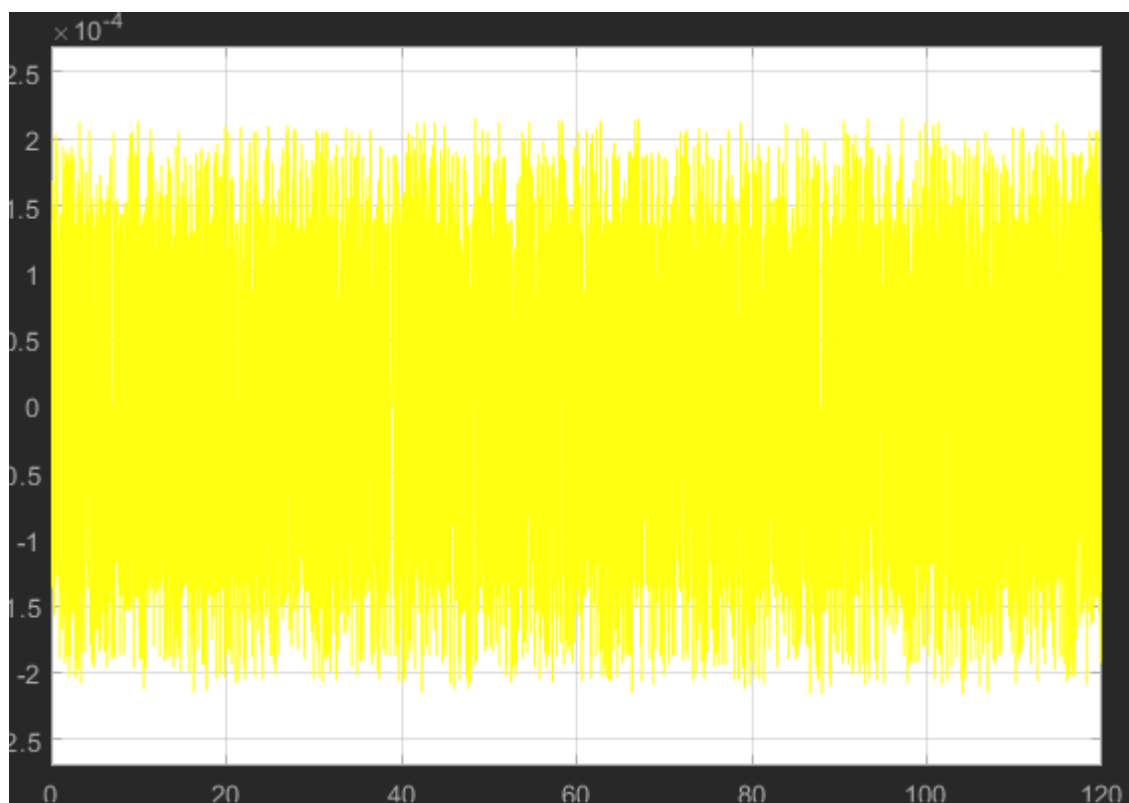


شکل 21 : زاویه ی پاندول به ازای قطب تند

سپس قطب های کند را قرار داده و خروجی موقعیت ارا به و زاویه را مشاهده می کنیم :



شکل 22 : موقعیت ارا به ازای قطب کند



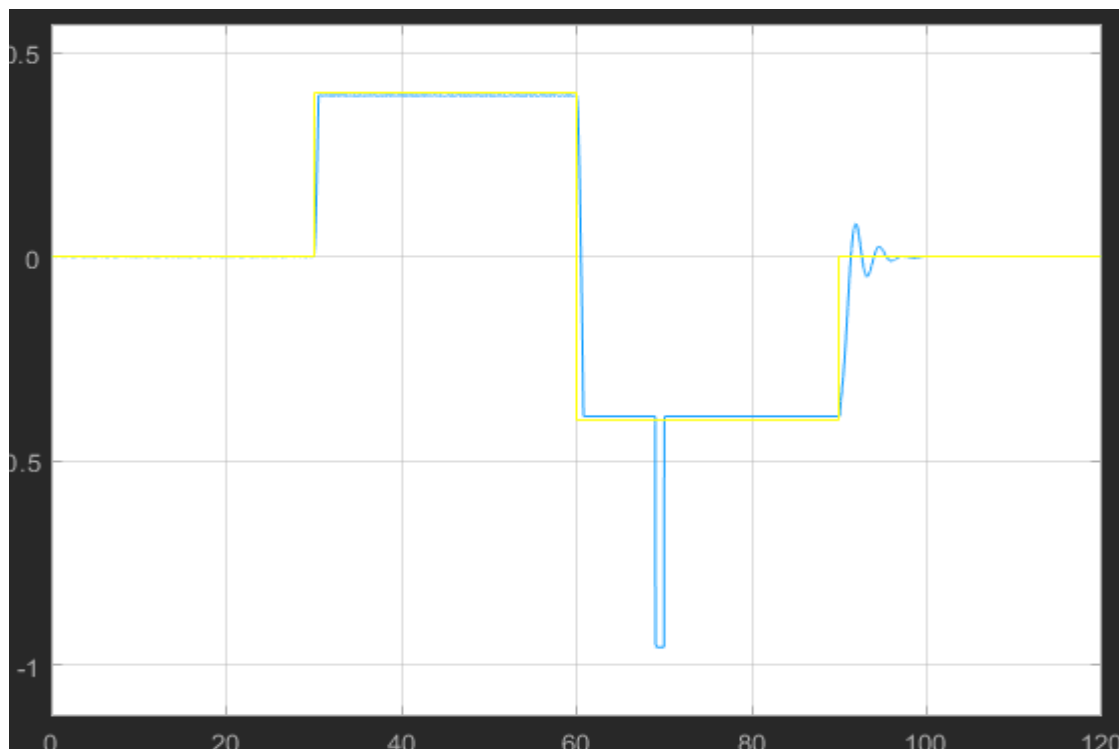
شکل 23 : زاویه ی پاندول به ازای قطب کند

مشاهده می کنیم به ازای قطب تند , مکان را تا حد مطلوبی توانسته ایم کنترل کنیم و فقط کمی بالازدگی داریم , زاویه را به طور کامل در همه زمان ها نتوانسته ایم کنترل کنیم و در حالاتی که مکان ارا به تغییر می کند کمی فرصت نیاز داریم تا زاویه به مقدار پایدار خود ($\varphi = 0$) برسد .

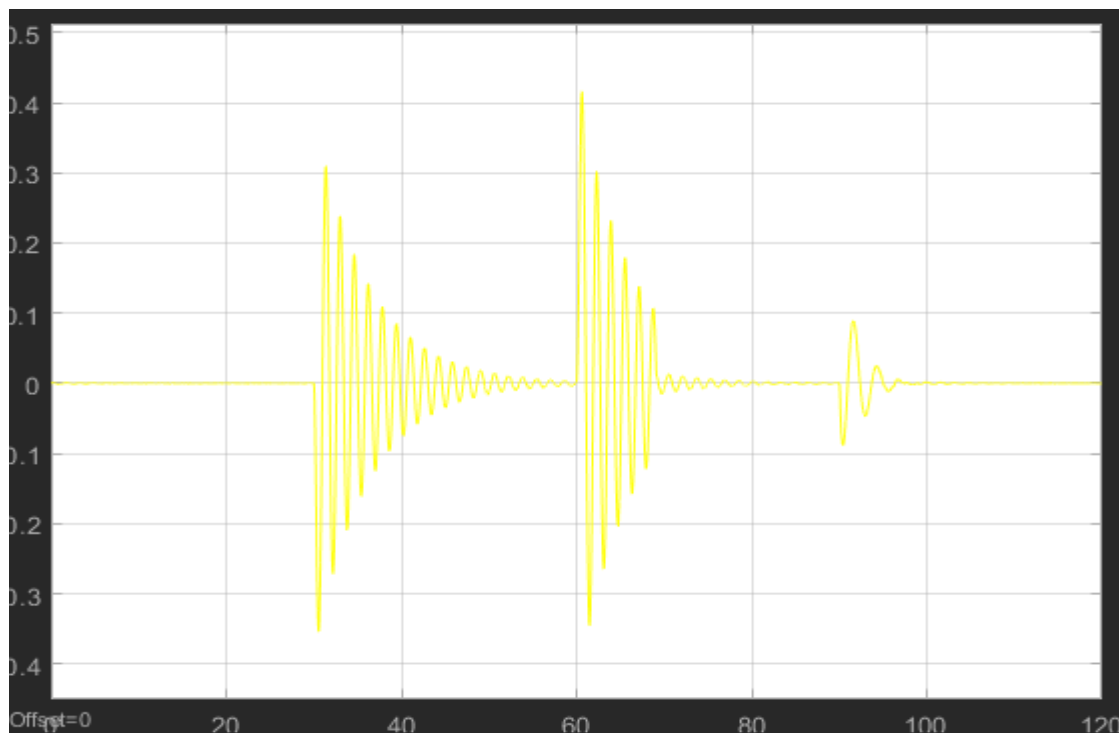
همچنین در حالتی که از قطب کند استفاده می کنیم , می بینیم که سیستم به درستی کنترل نشده و به مقادیر دلخواه نرسیده ایم .

می دانیم سیستم خطی را به ازای هر قطب مطلوبی می توان کنترل کرد و قابل ردیابی است زیرا می دانیم کنترلی که برای سیستم خطی طراحی کردیم حول نقطه تعادل بود و در واقع خطی سازی حول نقطه ی تعادل صورت گرفته است اما برای کنترل سیستم غیر خطی در نقطه تعادل قرار نداریم و همان طور که مشاهده کردیم ممکن است به ازای هر قطب دلخواهی در سیستم غیر خطی به جواب مطلوب نرسیم .

یک اغتشاش با اندازه ی 1 به مدت یک ثانیه در لحظه ی $t = 69$ به سیستم اعمال می کنیم و به ازای قطب های تند خروجی مکان و زاویه را مشاهده می کنیم :



شکل 24: موقعیت ارباب به ازای قطب کند به همراه اغتشاش به مدت یک ثانیه در لحظه $t = 69$



شکل 25: زاویه ی پاندول به ازای قطب کند به همراه اغتشاش به مدت یک ثانیه در لحظه $t = 69$

همان طور که مشاهده کردیم به ازای اعمال قطب تند ، سیستم مورد نظر نسبت به اغتشاش لحظه ای مقاوم بوده و مکان ارا به خوبی کنترل می شود .

اما اگر اغتشاش ثابت به سیستم بدهیم ، دیگر نسبت به اغتشاش مقاوم نخواهیم بود و باید از ردیاب با فیدبک حالت و کنترل انتگرال استفاده کنیم .

در این قسمت دامنه اغتشاش وارد شده به سیستم 1 بود و سیستم برای از بین بردن این اغتشاش به ورودی های بزرگ تر از حد اشباع نیاز نداشت و توانست به خوبی اغتشاش را برطرف کند . اما اگر دامنه اغتشاش را بزرگ تر کنیم ، سیستم برای برطرف کردن اغتشاش موجود نیاز به دامنه ورودی بزرگتری خواهد داشت که عملاً وارد ناحیه اشباع خواهیم شد ، در نتیجه یا به طور کلی سیستم نمی تواند اثر اغتشاش موجود را از بین ببرد یا خیلی دیر از بین خواهد برد .

-4

همان گونه که دیدیم ، کنترلر سیستم غیر خطی را با استفاده از ماتریس های بدست آمده از سیستم خطی طراحی کردیم . همچنین سیستم خطی را نیز حول نقطه تعادل خطی سازی کردیم . از فاز یک به خاطر داریم نقطه تعادل سیستم در مکان $x = x_1$ و زاویه ی صفر $\varphi = 0$ رخ می داد .

از آن جایی که کنترلر سیستم غیر خطی بر مبنای سیستم خطی ، طراحی شده است ، در نتیجه سیستم غیر خطی نیز سعی می کند سیستم را حول نقطه ی تعادل کنترل کند . در نتیجه با تغییر زاویه و فاصله گرفتن سیستم از نقطه تعادل خود ، برای این که سیستم بتواند از ورودی موقعیت ارا به پیروی کند سعی می کند زاویه خود را نیز در موقعیت $\varphi = 0$ حفظ کند . پس نتیجه می گیریم اگر سیستم غیر خطی پایدار باشد و قابل کنترل باشد ، خروجی ارا به مقدار مطلوب ما خواهد رسید .