به نام خدا

K. N. Toosi University of Technology

دانشگاه صنعتی خواجه نصیرالدین طوسی

دانشکده برق

کنترل مدرن

پروژه

[شهاب مقدادی نیشابوری]

[۴۰۰۰۹۴۴۳]

استاد : آقای دکتر تقی راد

فهرست مطالب

عنوان شماره صفحه

سوال اول تمرین دوم

ابتدا معادلات سیستم را به کمک مقاله ① به دست می آوریم:

- ورودی سیستم زاویه Elevator می باشد که آن را با δ_e نمایش می دهیم

- متغیرهای حالت سیستم شامل: $x=[u\ \alpha\ q\ \theta]$ می باشد که در آن:

* u، سرعت هواپیما در امتداد محور بدنه.

* α، زاویه حمله هواپیما (زاویه بین trajectory هواپیما و محور طولی هواپیما).

* θ، زاویه بین محور طولی هواپیما و محور x.

* q، مقدار q که برابر با سرعت زاویه ای هواپیما (pitch rate) است و از رابطه $q=\dot{\theta}$ به دست می آید.

معادلات خطی سازی نشده حالت، در صفحه ۲ مقاله ② قابل مشاهده است:

$$
\begin{aligned}
\dot{V} &= \frac{\bar{q}S\bar{c}q}{2mV}\left[C_{xq}(\alpha)\cos\alpha + C_{zq}(\alpha)\sin\alpha\right] \\
&+ \frac{\bar{q}S}{m}\left[C_x(\alpha,\delta_e)\cos\alpha + C_z(\alpha,\delta_e)\sin\alpha\right] \\
&- g\sin(\theta-\alpha) + \frac{T}{m}\cos(\alpha) \\
\dot{\alpha} &= q\left[1 + \frac{\bar{q}S\bar{c}}{2mV^2}\left(C_{zq}(\alpha)\cos\alpha - C_{xq}\sin\alpha\right)\right] \\
&+ \frac{\bar{q}S}{mV}\left[C_z(\alpha,\delta_e)\cos\alpha - C_x(\alpha,\delta_e)\sin\alpha\right] \\
&+ \frac{g}{V}\cos(\theta-\alpha) - \frac{T}{mV}\sin(\alpha) \\
\dot{\theta} &= q \\
\dot{q} &= \frac{\bar{q}S\bar{c}q}{2I_yV}\left[\bar{c}C_{mq}(\alpha) + \Delta C_{zq}(\alpha)\right] \\
&+ \frac{\bar{q}S\bar{c}}{I_y}\left[C_m(\alpha,\delta_e) + \frac{\Delta}{\bar{c}}C_z(\alpha,\delta_e)\right]
\end{aligned}
$$

در معادلات بالا، T، نیروی پیشران هواپیما و V همان u (سرعت هواپیما) می باشد.

که پس از ساده سازی و خطی سازی به فرم :

$$A_n = \begin{bmatrix} X_u & X_\alpha & 0 & -g_0 \cos\theta_0 \, \theta \\ \frac{Z_u}{u_0} & \frac{Z_\alpha}{u_0} & \frac{Z_q + u_0}{u_0} & \frac{-g_0 \sin\theta_0 \, \theta}{u_0} \\ M_u & M_\alpha & M_q & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

تبدیل می شود، پس از جایگذاری ثابت ها از مقاله ③ در صفحه 6 مقاله ① ماتریس A به فرم زیر تبدیل می شود:

$$A = \begin{bmatrix} -0.600896 & 251.751 & 0 & -9.815 \\ -0.0240377 & -11.9318 & 0.454903 & 0 \\ 0.00245226 & -0.616477 & -0.70952 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

خطی سازی ها با در نظر گرفتن α و θ در نزدیکی 0 و با فرض بسیار کوچک بودن $\theta\alpha$ انجام شده اند

سوال اول تمرین سوم

باتوجه به آنکه داریم :

$$e^{At} = I + At + \frac{1}{2!}A^2t^2 + \frac{1}{3!}A^3t^3 + \dots$$

ماتریس انتقال حالت به کمک برنامه متلب زیر به دست می آید :

```
A = [-0.600896 251.751 0 -9.815; -0.0240377 -11.9318 0.454904 0; 0.00245226 -0.616477 -0.70952 0; 0 0 1 0];
syms t
eAt = eye(4)
for i = 1:10
    eAt = eAt + (A^i)*(t^i)/(factorial(i));
end
```

(ماتریس حاصل به علت ابعاد زیاد، در فایل وورد آورده نشده، اما با اجرای eAt.m به راحتی قابل مشاهده است)

باتوجه به آن که داریم : (طبق 1)

$$B = \begin{bmatrix} 0 \\ -0.6951 \\ -9.9374 \\ 0 \end{bmatrix}$$

برنامه متلب زیر، پاسخ را برای دو شرط اولیه، $\underset{u=0}{\begin{bmatrix} 100 \\ 0 \\ 0 \\ 0 \end{bmatrix}}$ و $\underset{u=1}{\begin{bmatrix} 100 \\ 0 \\ 0 \\ 0 \end{bmatrix}}$ در $t=2$ به دست می آوریم.

```
T = 0:0.01:2;
B = [0;-0.6951;-1.9374; 0];
C = [1 0 0 0; 0 1 0 0; 0 0 1 0; 0 0 0 1];
D = [0];
sys = ss(A,B,C,D)
x0_1 = [100;0;0;0]
x0_2 = [100;1;0;0];
u2 = ones([201 1]);
u1 = zeros([201 1]);
[y1,tOut1] = lsim(sys,u1,T, x0_1);
[y2,tOut2] = lsim(sys,u2,T, x0_2);
```

```
>> y1

y1 =

  100.0000         0         0         0
   99.3719   -0.0226    0.0025    0.0000
   98.6944   -0.0425    0.0051    0.0001
   97.9741   -0.0599    0.0078    0.0001
   97.2170   -0.0753    0.0105    0.0002
   96.4284   -0.0887    0.0133    0.0003
   95.6130   -0.1004    0.0162    0.0005
   94.7750   -0.1106    0.0190    0.0006
   93.9181   -0.1195    0.0219    0.0009
   93.0457   -0.1271    0.0248    0.0011
   92.1606   -0.1337    0.0277    0.0013
   91.2656   -0.1393    0.0306    0.0016
   90.3629   -0.1441    0.0334    0.0020
   89.4545   -0.1481    0.0363    0.0023
   88.5423   -0.1514    0.0391    0.0027
   87.6279   -0.1542    0.0420    0.0031
   86.7128   -0.1564    0.0447    0.0035
   85.7980   -0.1582    0.0475    0.0040
   84.8849   -0.1595    0.0502    0.0045
```

y2 =

100.0000	1.0000	0	0
101.7291	0.8581	-0.0225	-0.0001
103.1115	0.7316	-0.0441	-0.0004
104.1861	0.6191	-0.0647	-0.0010
104.9875	0.5189	-0.0844	-0.0017
105.5469	0.4297	-0.1035	-0.0027
105.8918	0.3504	-0.1219	-0.0038
106.0469	0.2798	-0.1396	-0.0051
106.0342	0.2171	-0.1569	-0.0066
105.8733	0.1614	-0.1737	-0.0082
105.5817	0.1120	-0.1900	-0.0101
105.1752	0.0681	-0.2059	-0.0120
104.6674	0.0292	-0.2215	-0.0142
104.0710	-0.0053	-0.2368	-0.0165
103.3969	-0.0358	-0.2517	-0.0189
102.6551	-0.0628	-0.2664	-0.0215
101.8542	-0.0866	-0.2809	-0.0242
101.0022	-0.1076	-0.2951	-0.0271
100.1060	-0.1262	-0.3092	-0.0302
99.1718	-0.1424	-0.3231	-0.0333

همانطور که مشاهده می شود، درحالتی که هواپیما در وضعیت کاملا افقی قرار داشته باشد و ورودی ای برابر با ۵ به آن وارد شود، سرعت هواپیما کاهش پیدا کرده و همچنین دماغه هواپیما به طرف پایین خم می شود، که پس از شبیه سازی حالت دوم، مشاهده می شود شرایط اولیه دوم نیز، به همین اتفاق منجر می شود.

سوال اول تمرین ۱۴ام

برای تعیین رویت پذیر یا کنترل پذیر بودن سیستم، به کمک دستور obsv و ctrb،
ماتریس های کنترل پذیری و رویت پذیری سیستم را تشکیل داده، و بعد از آن rank آنها را
حساب می کنیم، در صورت full-rank بودن آنها، سیستم کنترل پذیر یا رویت پذیر است
همانطور که مشاهده می شود، سیستم هم کنترل پذیر و هم رویت پذیر می باشد

```
O =

   1.0e+04 *

    0.0001         0         0         0
         0    0.0001         0         0
         0         0    0.0001         0
         0         0         0    0.0001
   -0.0001    0.0252         0   -0.0010
   -0.0000   -0.0012    0.0000         0
    0.0000   -0.0001   -0.0001         0
         0         0    0.0001         0
   -0.0006   -0.3155    0.0105    0.0006
    0.0000    0.0136   -0.0006    0.0000
    0.0000    0.0008    0.0000   -0.0000
    0.0000   -0.0001   -0.0001         0
    0.0080    3.6149   -0.1504    0.0056
   -0.0003   -0.1543    0.0066   -0.0003
   -0.0000   -0.0098    0.0004   -0.0000
    0.0000    0.0008    0.0000   -0.0000
```

```
rnkO =

     4

Con =

   1.0e+04 *

         0   -0.0175    0.1990   -2.2214
   -0.0001    0.0007   -0.0083    0.0945
   -0.0002    0.0002   -0.0006    0.0061
         0   -0.0002    0.0002   -0.0006

rnkC =

     4
```

به کمک دستور jordan، ماتریس تبدیل T به دست می آید، پس داریم:

```
[T, J] = jordan(A);
JB = inv(T)*B;
JC = C*T;
```

```
J =

 -11.3393 + 0.0000i    0.0000 + 0.0000i    0.0000 + 0.0000i    0.0000 + 0.0000i
   0.0000 + 0.0000i   -1.2958 - 0.0000i    0.0000 + 0.0000i    0.0000 + 0.0000i
   0.0000 + 0.0000i    0.0000 + 0.0000i   -0.5539 - 0.0000i    0.0000 + 0.0000i
   0.0000 + 0.0000i    0.0000 + 0.0000i    0.0000 + 0.0000i   -0.0532 + 0.0000i
```

```
T =

   1.0e+03 *

   4.1936 + 0.0000i   0.1887 + 0.0000i  -0.0266 + 0.0000i  -0.0098 + 0.0000i
  -0.1788 + 0.0000i  -0.0005 - 0.0000i   0.0000 - 0.0000i   0.0000 - 0.0000i
  -0.0113 + 0.0000i  -0.0013 - 0.0000i  -0.0006 - 0.0000i  -0.0001 + 0.0000i
   0.0010 + 0.0000i   0.0010 + 0.0000i   0.0010 + 0.0000i   0.0010 + 0.0000i
```

```
JB =

    0.0036 - 0.0000i
    0.2018 + 0.0000i
    3.2863 - 0.0000i
   -3.4917 + 0.0000i
```

```
JC =

   1.0e+03 *

   4.1936 + 0.0000i   0.1887 + 0.0000i  -0.0266 + 0.0000i  -0.0098 + 0.0000i
  -0.1788 + 0.0000i  -0.0005 - 0.0000i   0.0000 - 0.0000i   0.0000 - 0.0000i
  -0.0113 + 0.0000i  -0.0013 - 0.0000i  -0.0006 - 0.0000i  -0.0001 + 0.0000i
   0.0010 + 0.0000i   0.0010 + 0.0000i   0.0010 + 0.0000i   0.0010 + 0.0000i
```

تحریک نشدن فرکانسی از سیستم، به آن معنی است که مود دینامیکی مرتبط با آن تحریک نشود. اگر ورودی u باشد، برای تحریک نشدن یک مود دینامیکی، نیاز است بردار شرایط اولیه، عمود بر بردار ویژه متناظر با آن مود باشد. بنابر این برای ساخت بردار عمود بر هر بردار ویژه.

$$v = \begin{bmatrix} x \\ y \\ z \\ k \end{bmatrix} \longrightarrow v' = \begin{bmatrix} y \\ -x \\ k \\ -z \end{bmatrix}$$

سوال اول تمرین پنجم

تابع تبدیل سیستم با روابط: $H = C * (SI-A)^{-1} * B$ بدست می‌آید

```
A = [-0.600896 251.751 0 -9.815; -0.0240377 -11.9318 0.454904 0; 0.00245226 -0.616477 -0.70952 0; 0 0 1 0];
B = [0;-0.6951;-1.9374; 0];
C = [1 0 0 0; 0 1 0 0; 0 0 1 0; 0 0 0 1];
D = [0];
Q = eye(4);
% syms s;
% H = C * inv(s * eye(4) - A) * B;
sys = ss(A,B,C,D)
tf(sys);
```

```
ans =

  From input to output...
                 -175 s^2 - 327 s + 222.7
   1:  ---------------------------------------------
       s^4 + 13.24 s^3 + 22.39 s^2 + 9.293 s + 0.4326

         -0.6951 s^3 - 1.792 s^2 - 0.8259 s - 0.4738
   2:  ---------------------------------------------
       s^4 + 13.24 s^3 + 22.39 s^2 + 9.293 s + 0.4326

               -1.937 s^3 - 23.85 s^2 - 25.79 s
   3:  ---------------------------------------------
       s^4 + 13.24 s^3 + 22.39 s^2 + 9.293 s + 0.4326

                 -1.937 s^2 - 23.85 s - 25.79
   4:  ---------------------------------------------
       s^4 + 13.24 s^3 + 22.39 s^2 + 9.293 s + 0.4326

Continuous-time transfer function.
```

برای نوشتن تحقق کانونی کنترل کننده داریم :

$$A_c = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -a_0 & -a_1 & -a_2 & -a_3 \end{bmatrix} , \quad B_c = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} , \quad C_c = \begin{bmatrix} b_{10} & b_{11} & b_{12} & b_{13} \\ & \vdots & & \\ b_{40} & b_{41} & b_{42} & b_{43} \end{bmatrix}$$

پس :

```
Ac =

         0    1.0000         0         0
         0         0    1.0000         0
         0         0         0    1.0000
   -0.4326   -9.2930  -22.3900  -13.2400

Bc =

     0
     0
     0
     1

Cc =

  222.7000 -327.0000 -175.0000         0
   -0.4738   -0.8259   -1.7920   -0.6951
         0  -25.7900  -23.8500   -1.9370
  -25.7900  -23.8500   -1.9370         0
```

از آنجا که تمام مقادیر ویژه منفی اند، پس سیستم پایدار داخلی است، و با توجه به نبود مودهای پنهان سیستم پایدار BIBO نیز می باشد

```
>> eig(A)

ans =

  -11.3393
   -1.2958
   -0.5539
   -0.0532
```

با توجه به آن که مقادیر ویژه ماتریس P، بزرگتر یا مساوی صفر است، سیستم پایدار مجانبی می باشد.

```
 P = lyap(A', Q);
P =

   1.0e+03 *

    0.0005    0.0092    0.0060    0.0001
    0.0092    0.1920    0.0327   -0.0719
    0.0060    0.0327    1.8071    1.2668
    0.0001   -0.0719    1.2668    0.9905
```

```
>> eig(P)

ans =

   1.0e+03 *

    0.0000
    0.0307
    0.2294
    2.7298
```

سوال اول تمرین ششم

آ) هدف از این بخش، طراحی یک کنترلر با استفاده از فیدبک خروجی برای رسیدن به نتایج مطلوب است، در این بخش ما میخواهیم برای ۴ تابع تبدیل، از ورودی به هرکدام از ۴ خروجی، یک کنترلر PID طراحی کنیم. برای این کار از برنامه PID tuner متلب استفاده میکنیم.

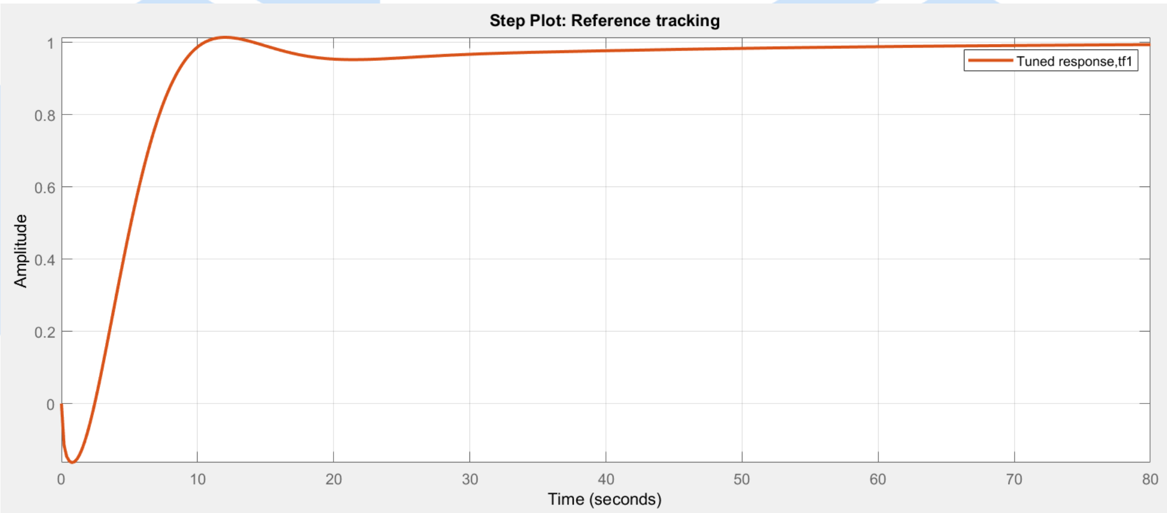
برای تابع تبدیل اول که مربوط به خروجی سرعت است، OS تا 5 درصد و زمان نشست تا ۴۰ ثانیه مناسب است، بنابر این با کمک متلب خواهیم داشت: (Ki ، Kp و Kd ضرایب PID میباشند)

Controller Parameters

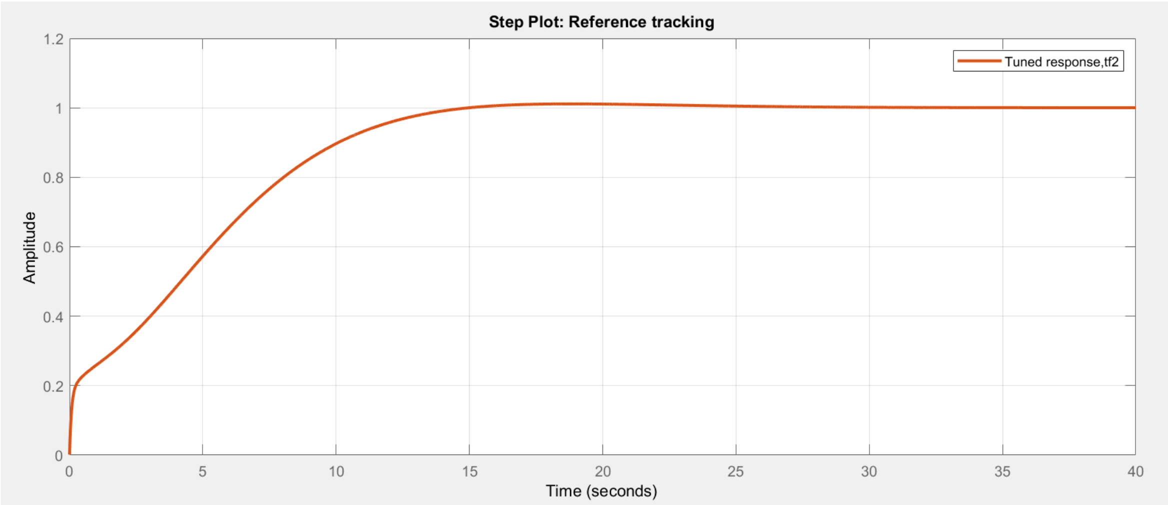
	Tuned
Kp	0.0075043
Ki	0.00027257
Kd	0.0069645
Tf	n/a

Performance and Robustness

	Tuned
Rise time	5.27 seconds
Settling time	39 seconds
Overshoot	1.46 %
Peak	1.01
Gain margin	10.8 dB @ 0.793 rad/s
Phase margin	60 deg @ 0.198 rad/s
Closed-loop stability	Stable



برای تابع تبدیل دوم که مربوط به زاویه حمله هواپیما است، نیاز است که فراجهش تا حد امکان پایین و نزدیک به ۱درصد باشد(به جهت حفظ پایداری هواپیما)، اما زمان نشست بالا مشکلی در عملکرد هواپیما بوجود نخواهد آورد، بنابراین خواهیم داشت:



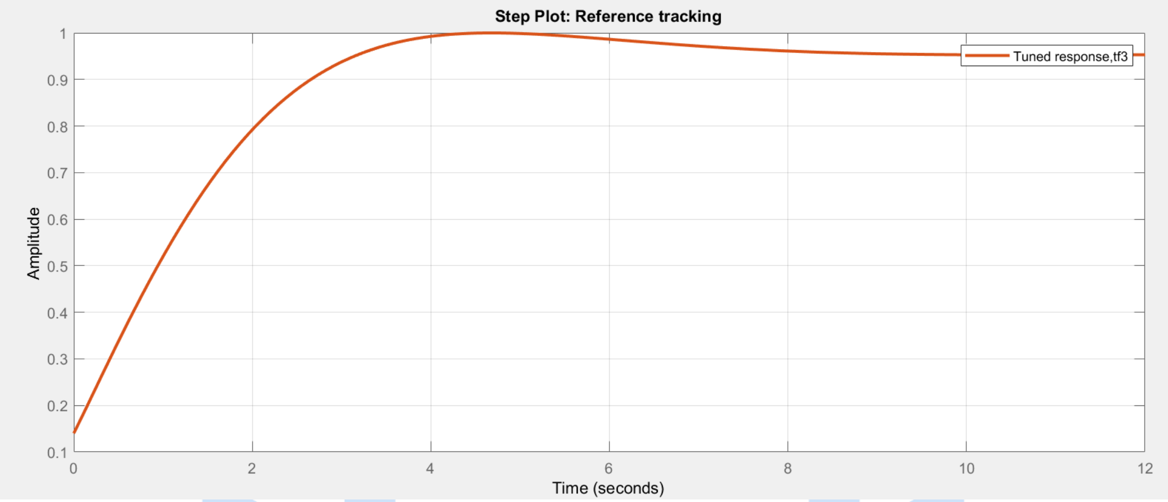
Controller Parameters

	Tuned
Kp	-3.8026
Ti	18.6124
Td	0.00052422
N	n/a

Performance and Robustness

	Tuned
Rise time	10 seconds
Settling time	13.2 seconds
Overshoot	1.14 %
Peak	1.01
Gain margin	Inf dB @ NaN rad/s
Phase margin	82 deg @ 0.191 rad/s
Closed-loop stability	Stable

برای متغیر حالت سوم که نشان دهنده سرعت زاویه ای است، میخواهیم فراجهش کمتر از ۵ درصد، و در عین حال، زمان نشست قابل قبولی داشته باشیم:



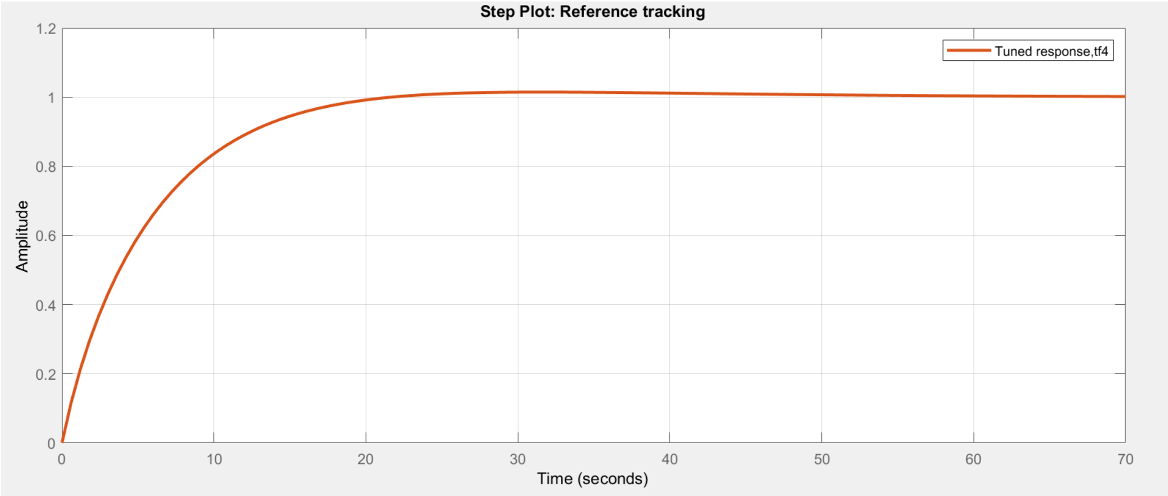
Controller Parameters

	Tuned
Kp	-0.34341
Ki	-0.35059
Kd	-0.084094
Tf	n/a

Performance and Robustness

	Tuned
Rise time	2.25 seconds
Settling time	7.08 seconds
Overshoot	4.77 %
Peak	1
Gain margin	Inf dB @ NaN rad/s
Phase margin	83 deg @ 0.759 rad/s
Closed-loop stability	Stable

برای متغیر چهارم نیز که pitch attitude هواپیما است، همانند angle of attack میخواهیم فراجهشی نزدیک به ۱ درصد و تا جای ممکن زمان نشست خوبی داشته باشد، بنابرین:



Controller Parameters

	Tuned
Kp	-0.060312
Ki	-0.0034663
Kd	-0.11325
Tf	n/a

Performance and Robustness

	Tuned
Rise time	11.9 seconds
Settling time	18.3 seconds
Overshoot	1.45 %
Peak	1.01
Gain margin	Inf dB @ NaN rad/s
Phase margin	90 deg @ 0.175 rad/s
Closed-loop stability	Stable

ب) با توجه به آن که در سیستم درنظر گرفته شده، خروجی برداری شامل تمام متغیر های حالت است، در این حالت، تفاوتی میان کنترلر I/O و کنترلر حالت وجود ندارد.

ج) قسمت حقیقی قطب های غالب (قطب هایی که به محور موهومی نزدیک ترند) میزان ضریب میرایی سیستم را تعیین می کند .هرچه ضریب میرایی بالاتر باشد (یعنی قطب ها از محور موهومی دورتر باشند) باعث overshoot کمتر و پاسخ نرم تری می شود .برعکس، هرچه ضریب میرایی پایین تر باشد (یعنی قطب ها به محور موهومی نزدیک تر باشند) باعث overshoot بیشتر و پاسخ نوسانی تری می گردد.

بنابرین با توجه به آنکه:

```
>> eig(A)

ans =

  -11.3393
   -1.2958
   -0.5539
   -0.0532
```

بنظر میرسد بهترین حالت جایابی قطب ها، دو قطب بر روی 0.5- و دو قطب بر روی 5- باشد.

با استفاده از دستور place متلب، مقادیر k برای فیدبک حالت به فرم:

```
k =

    0.0415    5.3063   -1.7845   -0.0729
```

به دست خواهد آمد. همچنین فرض میکنیم در این قسمت خروجی مد نظر تنها زاویه حمله هواپیما است، بنابراین:

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

$$-0.6009\,v + 251.7510 + \theta \times (-9.8150) = 0,$$
$$-0.024\,v - 11.9318 - 0.6951\,u = 0,$$
$$0.0025\,v - 0.6165 + u \times (-1.9374) = 0$$
$$=> \alpha^* = \alpha_d\,, q^* = 0, v^* = -470.36\,, \theta^* = 54.44\,, u* = -0.925$$

پس با توجه به آن که هدف رسیدن به زاویه حمله مطلوب(۱ درجه برای این بخش) است داریم:

$$u = u^* - k\Delta x$$

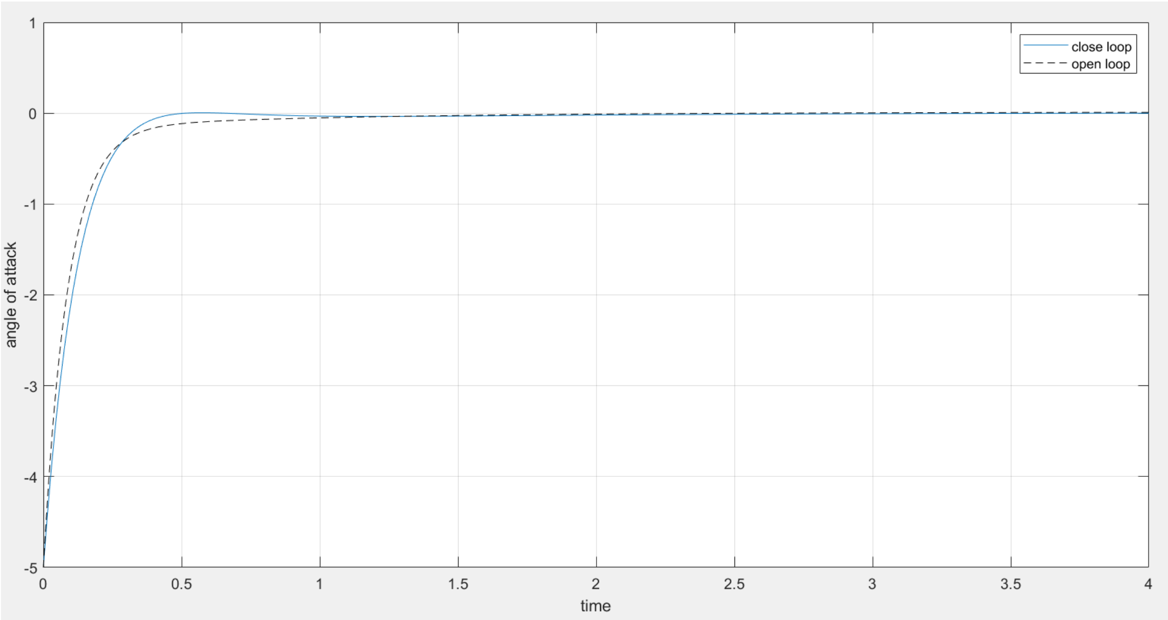
$$u = -0.925 - [0.0415 \quad 5.3063 \quad -1.7845 \quad -0.072](\begin{bmatrix} v \\ \alpha \\ q \\ \theta \end{bmatrix} - \begin{bmatrix} -470.36 \\ 1 \\ 0 \\ 54.44 \end{bmatrix})$$

اگر شرایط اولیه را به فرم:

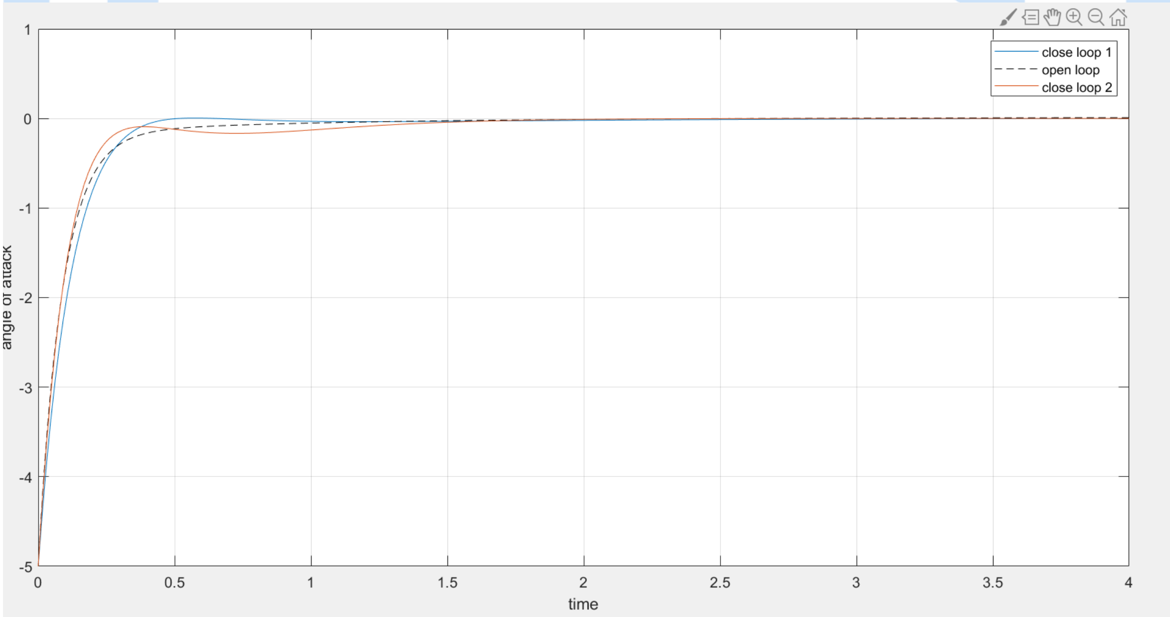
```
xx0 =

   200
    -5
     0
     0
```

یعنی سرعت هواپیما برابر ۲۰۰ متر بر ثانیه و زاویه حمله ابتدایی آن، -۱۵ درجه باشد، در این حالت، اگر هدف را رساندن زاویه حمله به ۰ درجه فرض کنیم، که باعث پایدار شدن وضعیت هواپیما میشود، سیستم حلقه بسته، با زمان بهتر و فراجهش نزدیک به ۰ درصد، میتواند وضعیت هواپیما را پایدار کند.



حال اگر محل قطب های سیستم را عوض کنیم، به گونه ای که قطب های روی 0.5- به 2.5- انتقال یابند، نتیجه زیر حاصل میشود، علت این نتیجه آن است که سیستم سریع تر شده، اما در عوض سریع تر شدن، دیر تر به حالت ماندگار خود که عدد ۱ است، میرسد. در کاربرد های متفاوت، هر دو حالت میتوانند مفید واقع شوند:

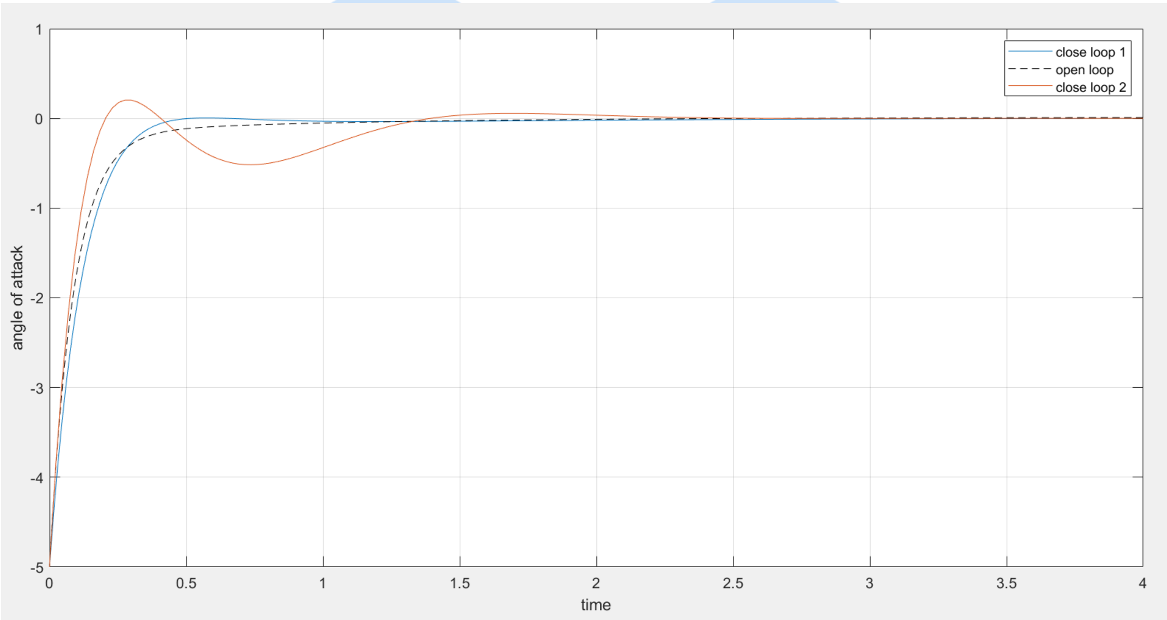


اما اگر سرعت سیسستم از اهمیت زیادی برخوردار باشد، میتوانیم تا مقدار ۵ درصد فراجهش را صرف نظر کرده و با جایابی قطب ها روی:

```
Pd2 =

  -3.0000 + 3.0000i  -3.0100 - 3.0000i  -4.0000 + 1.0000i  -4.0100 - 1.0000i
```

سرعت سیستم را بهبود ببخشیم:(فراجهش ۳.۵ درصد)



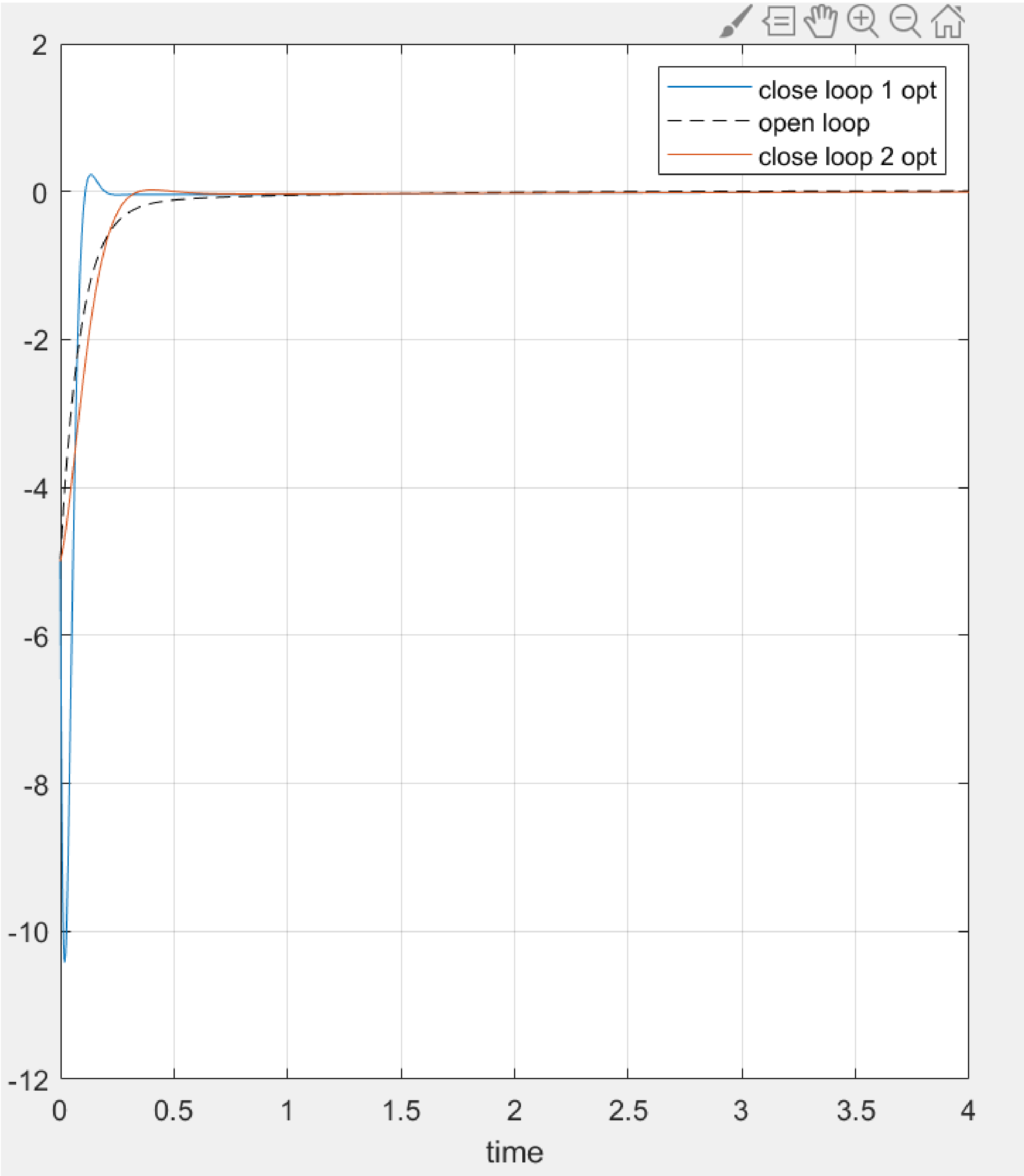
د) با توجه به قسمت (آ) میتوان گفت که تلاش کنترلی ما در این پروژه، به مراتب اهمیت کمتری از سریعتر به نتیجه رسیدن سیستم دارد، بنابر این R باید به مراتب کوچک تر از Q باشد. به همین دلیل R را برابر با ۱ و Q را یک ماتریس قطری با قطر ۱۰۰ و ۴*۴ در نظر میگیریم، با کمک دستور lqr متلب، k متناسب با شاخص های ما به صورت زیر به دست می آید:

```
k_opt =

   -3.5320   62.2860  -49.9622 -118.5632
```

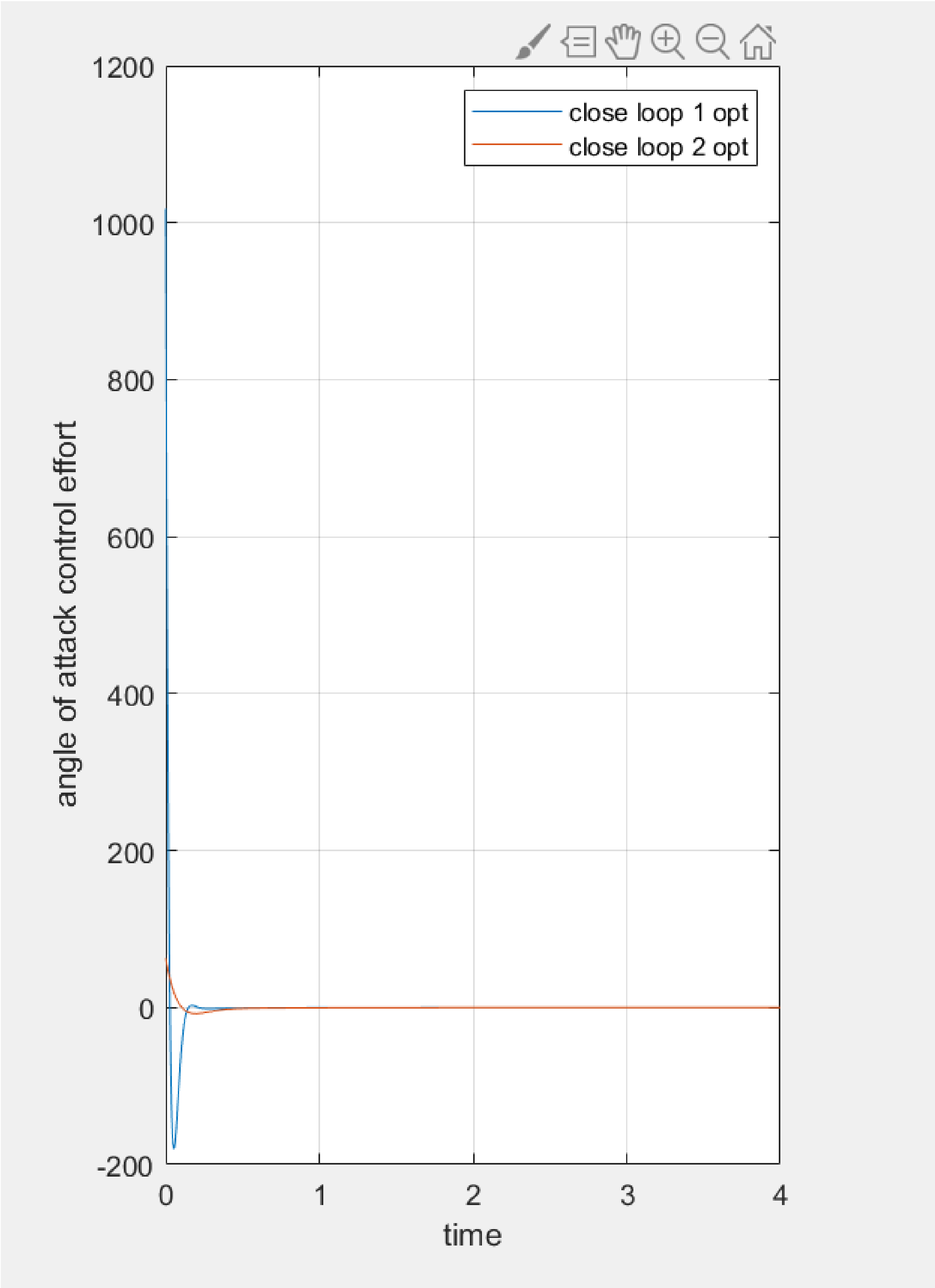
اما اگر به تلاش کنترلی نیز برای ما اهمیت داشته باشد، ماتریس Q را ماتریس همانی 4*4 و R را یک در نظر میگیریم.

در شکل زیر نمودار ۲ حالت بالا مقایسه شده:



همانطور که مشاهده میشود در حالت اول که تلاش کنترلی اهمیتی ندارد، سریع تر به جواب رسیدیم، اما تلاش کنترلی بیشتری به کار بردیم و overshoot بالاتری داشتیم، اما در قسمت دوم که مقادیر R و Q هم ارز بودند، میبینیم سرعت به جواب رسیدن سیستم، به مراتب کاهش داشته است.

تلاش کنترلی دو حالت بیان شده، به صورت زیر است:



همانطور که مشخص است، تلاش کنترلی در حالت اول به شدت زیاد، و با برابر شدن R و Q کمتر شده است.