



28.2.1395

# پروژه چهارم درس MPC

طراحی کنترل کننده PFC

نام دانشجو: فاطمه زاهدی

شماره دانشجویی: 94201583

نام استاد: دکتر حائری

## 1. معرفی سیستم مورد استفاده:

سیستم مورد استفاده در این شبیه سازی یک سیستم غیر خطی CSTR می باشد که معادلات فضای حالت آن به فرم زیر است.

$$\frac{dC_A}{dt} = \frac{q}{V}(C_{A0} - C_A) - k_o C_A \exp\left(\frac{-E}{RT}\right)$$

$$\frac{dT}{dt} = \frac{q}{V}(T_0 - T) - \left(\frac{-\Delta H}{\rho C_p}\right) k_o C_A \exp\left(\frac{-E}{RT}\right) + \left(\frac{\rho_c C_{pc}}{\rho C_p V}\right) q_c \left[ 1 - \exp\left(\frac{-hA}{q_c \rho_c C_{pc}}\right) \right] (T_{c0} - T)$$

نقطه کاری که در آن سیستم بالا را خطی سازی کرده ایم به صورت زیر می باشد.

$$C_A = 0.0882, \quad T = 441.2, \quad q = 100, \quad q_c = 100.$$

سیستم خطی سازی شده در نقطه کار مذکور و پس از جایگزینی پارامترها به صورت زیر درآمد.

$$A = \begin{pmatrix} -11.3 & 0 \\ 2064.8 & 7.4 \end{pmatrix}, \quad B = \begin{pmatrix} 0.0091 & 0 \\ -0.912 & -0.9053 \end{pmatrix}, \quad C = (0 \quad 1), \quad D = (0 \quad 0).$$

## 2. طراحی کنترل کننده PFC:

برای طراحی کنترل کننده PFC از روش شناسایی سیستم استفاده شده است. بدین منظور از روش ARMARKOV استفاده کرده ایم. که معادلات این روش در زیر آورده شده است.

$$y(t + \mu) = g_1^\mu \Delta u(t + \mu - 1) + g_2^\mu \Delta u(t + \mu - 2) + \dots + g_\mu^\mu \Delta u(t) + \beta_1^\mu \Delta u(t - 1) + \dots + \beta_n^\mu \Delta u(t - n) + \alpha_1^\mu y(t - 1) + \dots + \alpha_n^\mu y(t - n) = \varphi^T \theta_\mu$$

بنابراین ضرایب  $\alpha$ ,  $\beta$  و  $g$  را با استفاده از روش شناسایی بدست می آوریم. بدین صورت که دو ورودی رندم تولید می کنیم و به سیستم می دهیم و حاصل را در  $Y$  ذخیره می کنیم.

ما طراحی را به ازای دو  $\mu$  و یک حالت برای ورودی انجام دادیم. که  $\mu$ ها را برابر با 3 و 5 در نظر گرفتیم و تغییرات ورودی آینده  $\Delta u(t+2)$  را در نظر گرفتیم. از طرفی سایر پارامترها مانند  $P$ ,  $T_s$  را مشابه روشی که در پروژه های قبلی انجام می دادیم تعیین کردیم که مقدار آنها به ترتیب برابر با 0.1 و 5 شد. همچنین مقدار  $n$  را با توجه به مرتبه سیستم برابر با 3 قرار دادیم. لازم به ذکر است که در این طراحی می توان هم آینده و هم گذشته را از روش IIR بدست آورد یا اینکه می توان گذشته را از IIR و آینده را از FIR بدست آورد. کد مربوط به هر دو حالت نوشته شده است. بعد از شناسایی  $g$  ها و مقایسه با  $g$  هایی که از پاسخ پله بدست می آید مشاهده کردیم که  $g$ ها در هر دو روش تقریباً با هم برابرند که این نشان می دهد که این شناسایی به درستی صورت گرفته است. بنابراین با توجه به این مقایسه ماتریس toeplitz در هر دو حالت تقریباً مشابه هم شد و هر دو حالت ما را به یک پاسخ می رساند.

بعد از پیاده سازی این کنترل کننده پاسخ سیستم غیر خطی و قانون های کنترلی به ازای حالتی که خروجی مطلوب پله واحد به سیستم داده ایم به صورت آورده شده در شکل زیر در آمد. لازم به ذکر است که مقدار گاما برابر با 0.02 و مقدار آلفا برابر با نیم در نظر گرفته شده است.

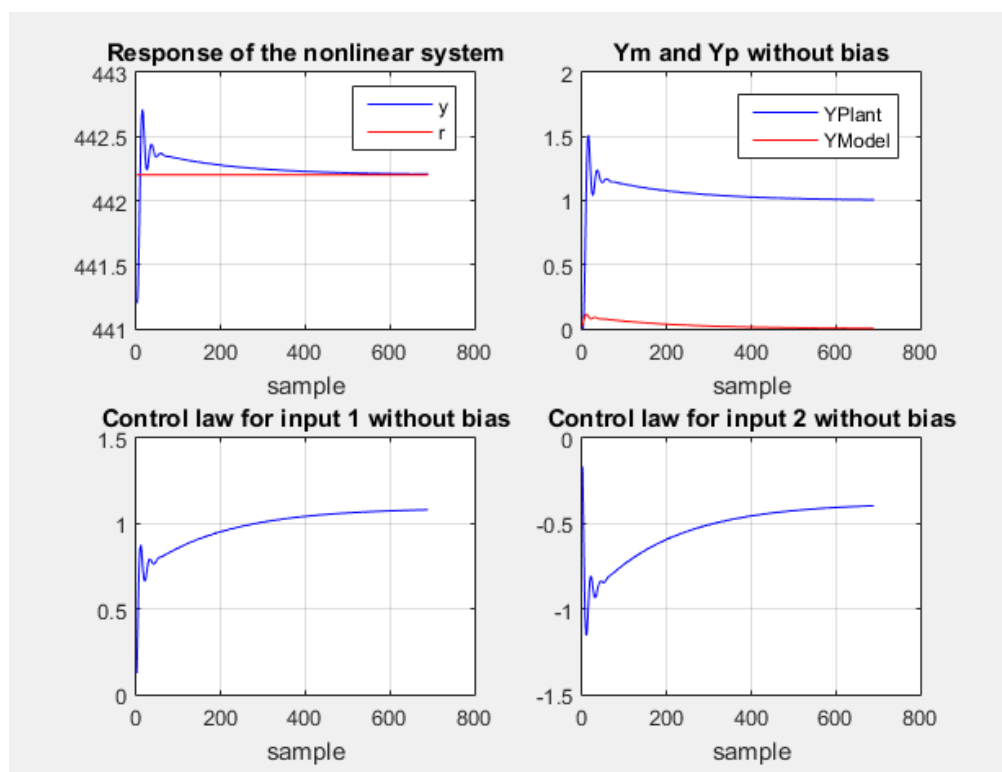


Figure1. PFC on nonlinear system.

### 3. پاسخ سیستم به ازای خروجی مطلوب های مختلف:

در این بخش خروجی های مطلوب مختلفی را به سیستم می دهیم و پاسخ سیستم را مشاهده می کنیم. لازم به ذکر است در کلیه قسمت های این بخش گاما برابر با 0.02 و آلفا برابر با 0.5 قرار داده شده است.

#### 3.1. خروجی مطلوب پالس:

در این قسمت به ازای خروجی مطلوب سیستم، پالس می دهیم و پاسخ سیستم به همراه قانون کنترلی بدست آمده از کنترل کننده و خروجی مدل به صورت زیر در آمد.

همانطور که می بینیم خروجی سیستم، خروجی مطلوب را دنبال می کند.

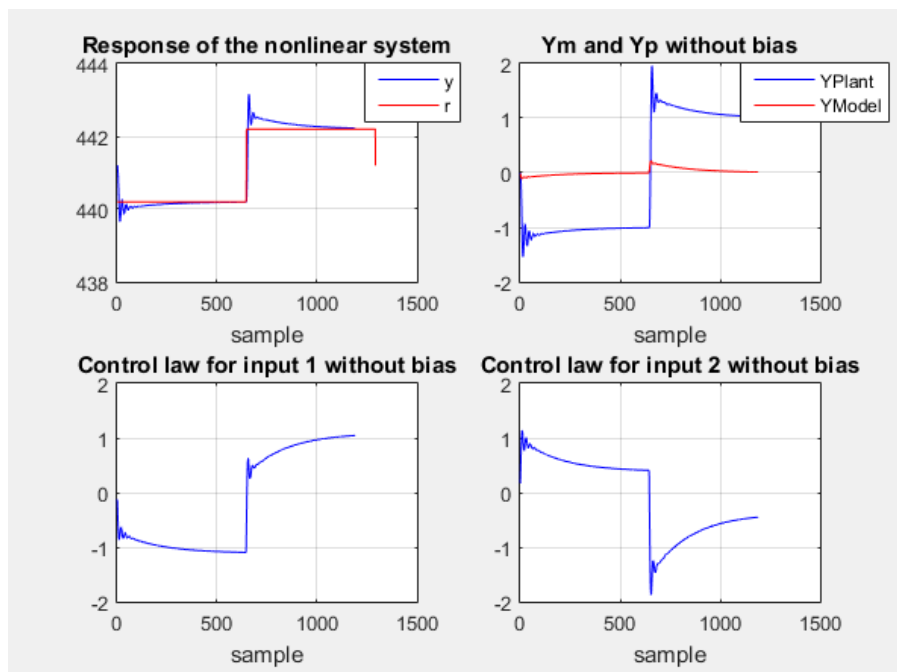


Figure 2. Response of the system with pulse as a desired output.

### 3.2. خروجی مطلوب سینوسی:

در این قسمت به ازای خروجی مطلوب سیستم، سیگنال سینوسی می‌دهیم و پاسخ سیستم به همراه قانون کنترلی بدست آمده از کنترل کننده و خروجی مدل به صورت زیر در آمد.

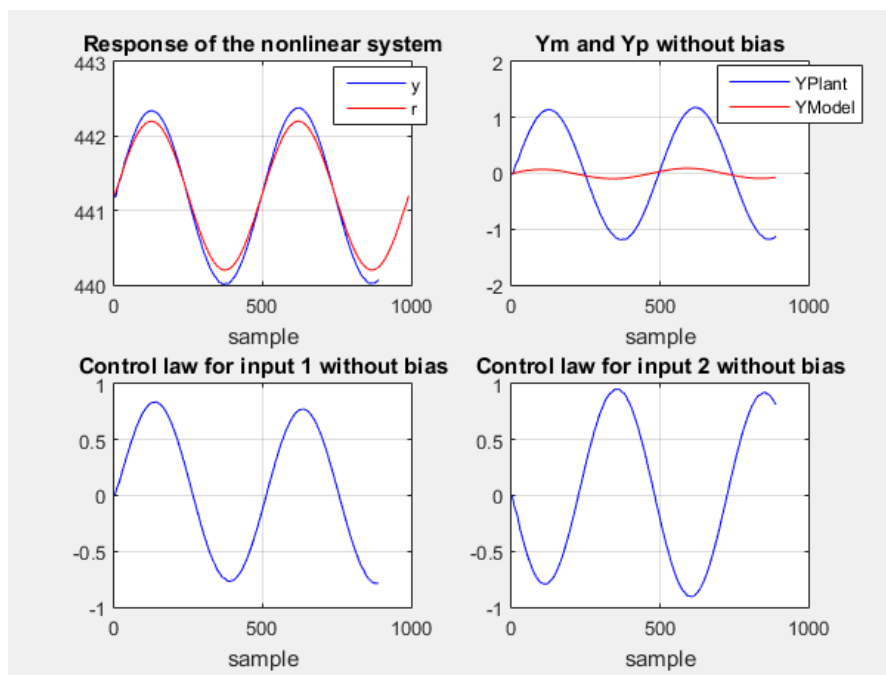


Figure 3. Response of the system with sine as a desired output.

همانطور که در شکل بالا می‌بینیم پاسخ سیستم، خروجی مطلوب را دنبال می‌کند. قوانین کنترلی و خروجی مدل نیز در این شکل قرار داده شده است و می‌توان نحوه رفتار آنها را دید.

### 3.3. خروجی مطلوب پله‌ای با پرش‌های مختلف:

در این قسمت به ازای خروجی مطلوب سیستم، استپ با پرش‌های متفاوت می‌دهیم و پاسخ سیستم به ازای این خروجی مطلوب به صورت زیر شد. همانطور که در شکل زیر می‌بینیم خروجی سیستم، خروجی مطلوب را دنبال می‌کند.

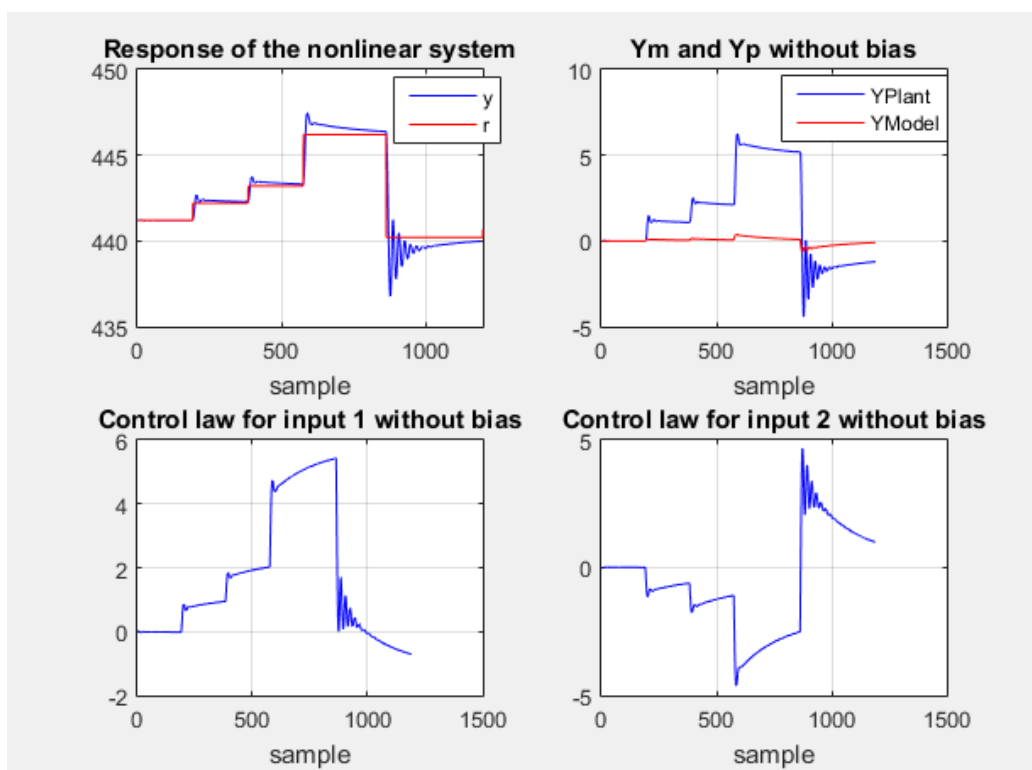


Figure 4. Response of the system with step with various jumps as a desired output.

### 3.4. خروجی مطلوب پله‌ای:

در این قسمت پاسخ سیستم به ازای خروجی مطلوب سیگنال پله‌ای را رسم می‌کنیم. این خروجی مطلوب را به ازای دامنه‌های مختلف شامل 0.1%، 0.25% و 1% نقطه کار رسم می‌کنیم. شکل زیر پاسخ را به ازای این خروجی مطلوب با دامنه 0.1% نقطه کار نشان می‌دهد. همانطور که در شکل زیر می‌بینیم پاسخ سیستم، خروجی مطلوب را دنبال می‌کند و در قسمت‌هایی که خروجی مطلوب صفر هست خروجی سیستم نیز صفر شده است.

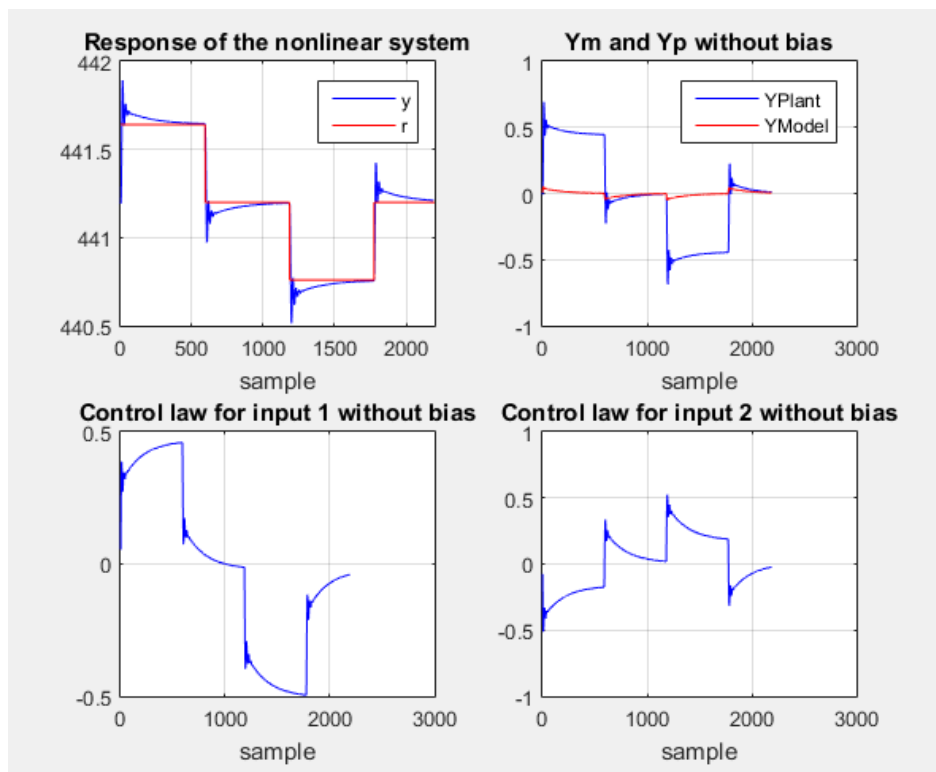


Figure 5. Response of the system with step with 0.1% of operating point as a desired output.

در شکل زیر پاسخ سیستم به ازای همین خروجی مطلوب اما به ازای دامنه 0.25% نقطه کار را می بینیم.

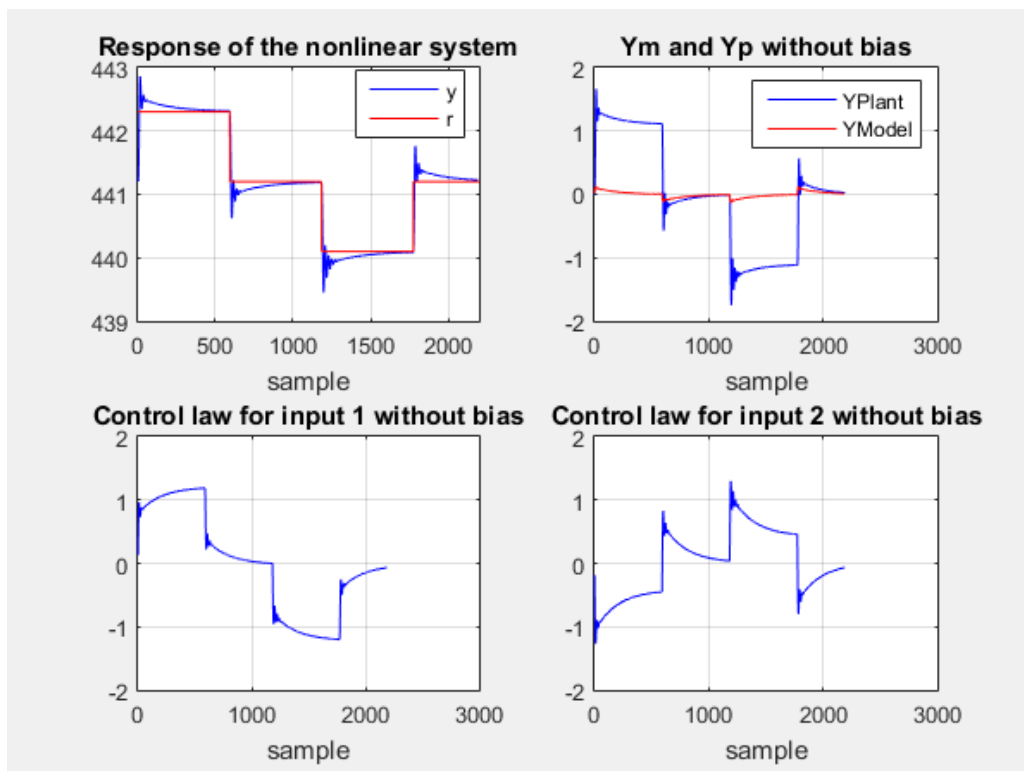


Figure 6. Response of the system with step with 0.25% of operating point as a desired output.

همانطور که در شکل بالا می‌بینیم پاسخ سیستم، خروجی مطلوب را دنبال می‌کند و در ناحیه‌هایی که خروجی مطلوب صفر است پاسخ سیستم نیز صفر است.

در شکل زیر پاسخ سیستم به ازای همین خروجی مطلوب اما به ازای دامنه 1% نقطه کار می‌بینیم.

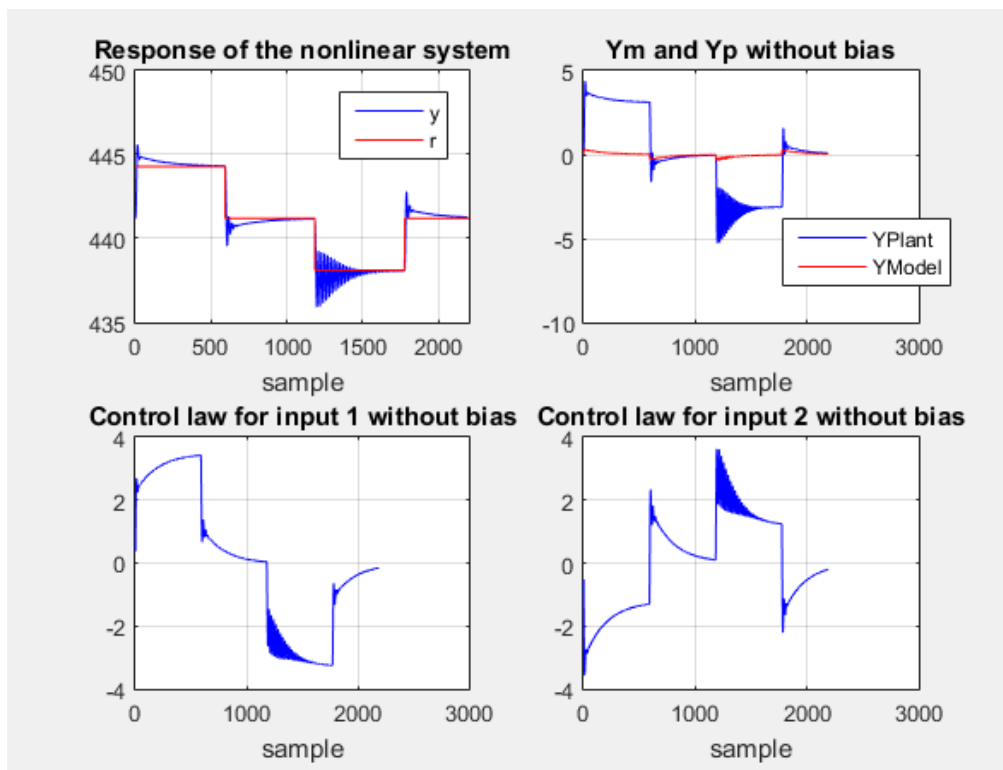


Figure 7. Response of the system with step with 1% of operating point as a desired output.

همانطور که در شکل بالا می‌بینیم پاسخ سیستم خروجی مطلوب را دنبال کرده است اما در ناحیه که دامنه استپ منفی است پاسخ مقداری به نوسان افتاده و خراب شده که در واقع اثر غیر خطی بودن سیستم است که خود را با افزایش دامنه استپ‌ها در پاسخ سیستم نشان داده است.

## 4. بررسی اثر پارامترها:

در این بخش تاثیر تغییرات پارامترهای مورد استفاده در طراحی کنترلر PFC را بر روی پاسخ سیستم بررسی می‌کنیم.

### 4.1. بررسی ماتریس R:

در این قسمت برای اینکه تاثیر تغییر ماتریس R را بر روی پاسخ سیستم ببینیم گاما را به ازای اعداد 1، 1/60 و 0.006 بررسی می‌کنیم و آلفا را در این قسمت ثابت و برابر با 0.5 قرار داده ایم. شکل زیر تاثیر گاما را به ازای اعداد مذکور نشان می‌دهد.

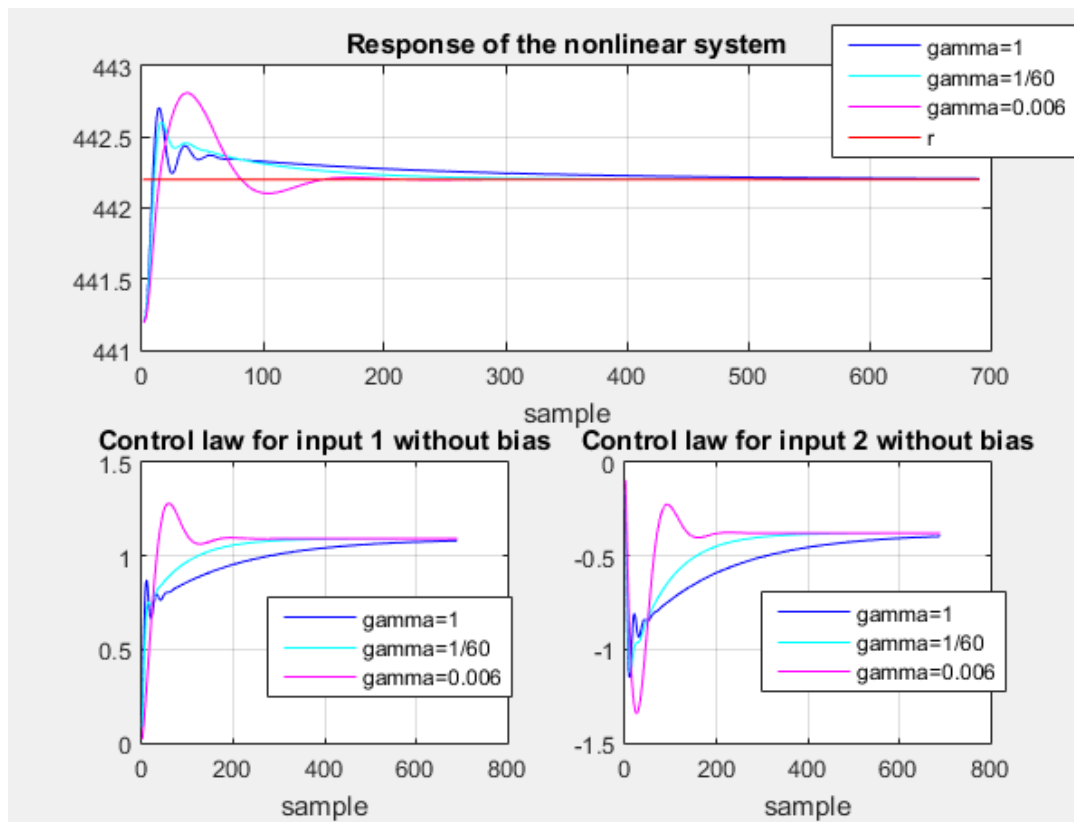


Figure 8. Responses of the system in verifying  $R$ .

همانطور که در شکل 8 می‌بینیم کاهش گاما تاثیری در حالت ماندگار پاسخ سیستم نمی‌گذارد و خطای پاسخ در حالت ماندگار همواره صفر است. اما همانطور که در این شکل می‌بینیم پاسخ سیستم به ازای کاهش گاما تندتر شده است.

## 4.2. بررسی ماتریس $Q$ :

در این قسمت به ازای گامای 1 و آلفای 0.5 پاسخ سیستم را به ازای  $Q=I$ ،  $Q=60I$  و  $Q=160I$  بررسی می‌کنیم.

شکل 9 تاثیر افزایش  $Q$  را به ازای مقادیر گفته شده نشان می‌دهد.

همانطور که در این شکل می‌توان دید با افزایش  $Q$  پاسخ سیستم تندتر می‌شود. اما دقیقاً مشابه کاهش گاما، خطای حالت ماندگار تغییری نمی‌کند و همواره صفر می‌ماند.



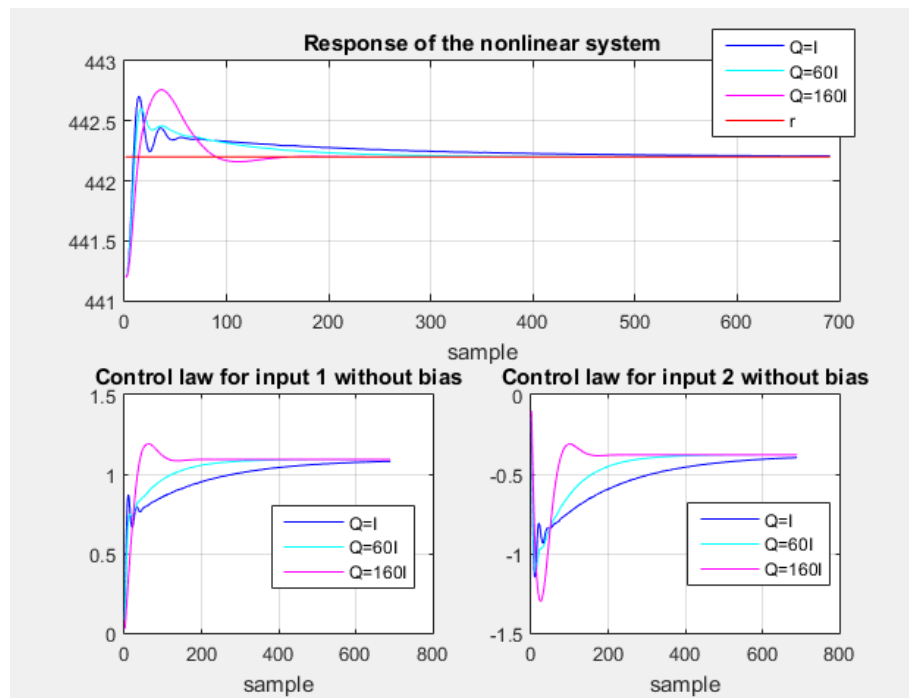


Figure 9. Responses of the system in verifying  $Q$ .

### 4.3. بررسی پارامتر آلفا:

در این قسمت تاثیر پارامتر آلفا را بر روی پاسخ سیستم بررسی می کنیم. شکل زیر پاسخ سیستم به ازای مقادیر آلفا 0.4، 0.6 و 0.8 نشان می دهد. در طی این بررسی مقدار گاما تقریباً 0.02 قرار داده شده است.

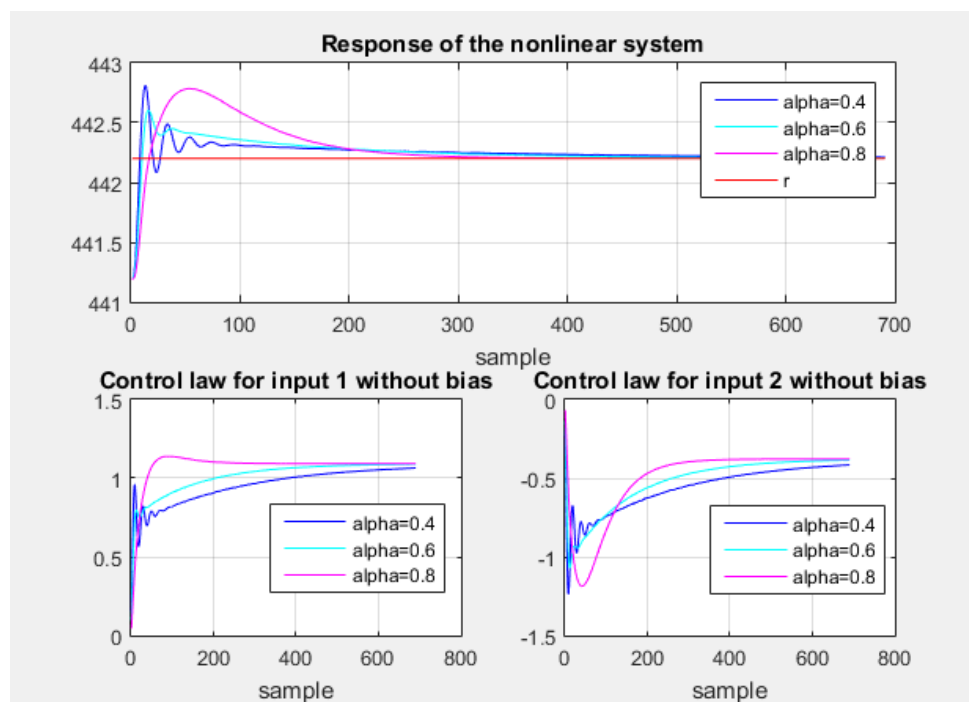


Figure 10. Responses of the system in verifying  $\alpha$ .

همانطور که در شکل بالا می‌بینیم کاهش آلفا باعث تند شدن پاسخ سیستم شده است.

#### 4.4. بررسی پارامتر $T_s$ :

در این بخش به بررسی اثر تغییرات  $T_s$  بر روی پاسخ سیستم می‌پردازیم. شکل زیر پاسخ سیستم به ازای  $T_s$  هایی برابر با 0.05، 0.1 و 0.5 نشان می‌دهد.

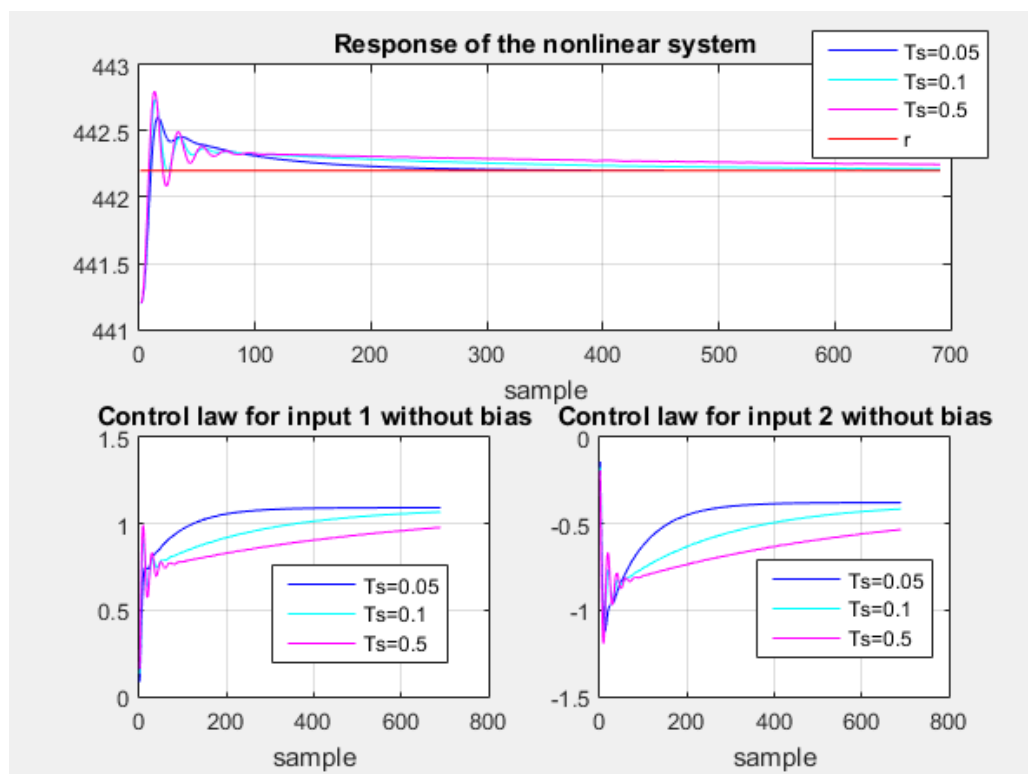


Figure 11. Responses of the system in verifying  $T_s$ .

همانطور که در شکل بالا می‌بینیم وقتی زمان نمونه برداری را کم می‌کنیم باعث تند شدن پاسخ سیستم می‌شود. اما این تند شدن پاسخ تا جایی ادامه دارد یعنی از یک مقداری به بعد دیگر کاهش  $T_s$  تاثیر چندانی در سریع شدن پاسخ ندارد.

#### 4.5. بررسی تاثیر فاصله بین دو $\mu$ از هم:

در این بخش  $\mu_1$  و  $\mu_2$  را تغییر می‌دهیم به طوری که فاصله آنها متفاوت باشد. یعنی این دو مقدار را طوری تعیین می‌کنیم که اختلاف آنها دو، سه و چهار باشد. بنابراین در حالت a این دو مقدار برابر با  $\mu_1=3$ ،  $\mu_2=5$  می‌باشد. در حالت b این دو مقدار برابر با  $\mu_1=3$ ،  $\mu_2=6$  است و در حالت c این دو مقدار برابر با  $\mu_1=3$ ،  $\mu_2=7$  می‌باشد. پاسخ به ازای این سه حالت برای دو مقدار  $\mu$  در شکل زیر آورده شده است.

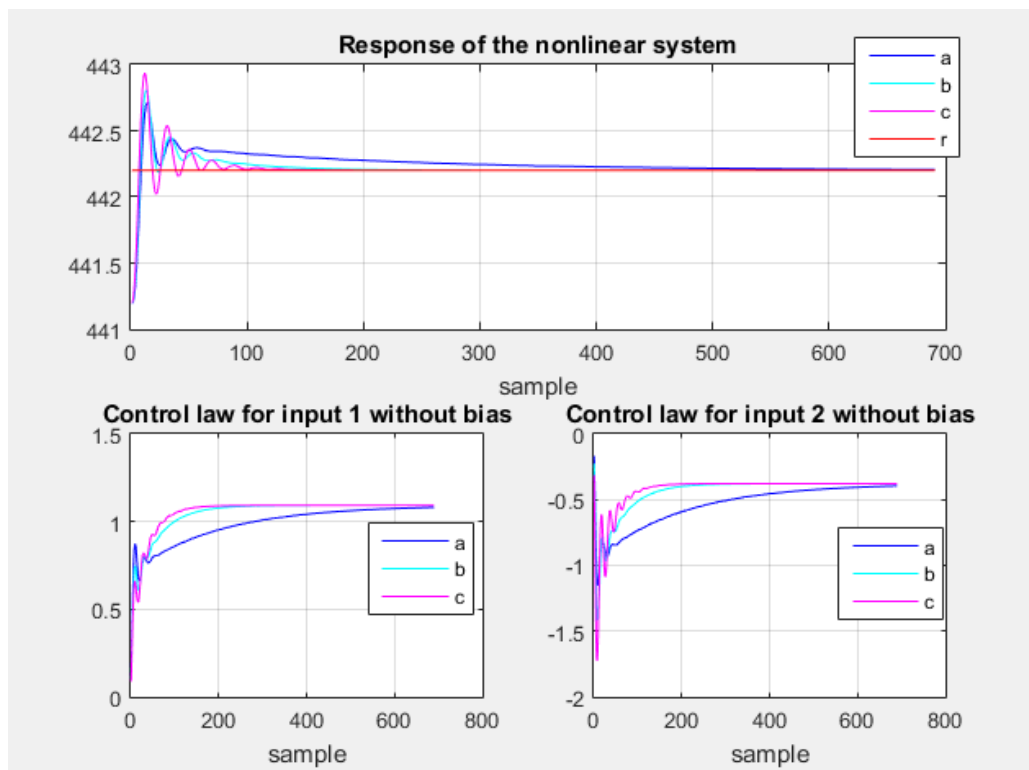


Figure 12. Responses of the system in verifying the difference between  $\mu s$ .

همانطور که در شکل بالا می‌بینیم با افزایش فاصله بین  $\mu$ ها پاسخ نوسانی‌تر شده اما سریع‌تر شده است.

## 5. بررسی اثر اغتشاش و نویز:

در این بخش روی خروجی پروسه نویز سفید و اغتشاش اضافه می‌کنیم و اثر اضاف شدن این دو را در پاسخ سیستم مشاهده می‌کنیم.

### 5.1. بررسی اثر اغتشاش:

با اضافه کردن اغتشاش با دامنه یک، پاسخ سیستم به صورتی که در شکل زیر آورده شده است درآمد. در این شکل اغتشاش نیز رسم شده است. همانطور که در این شکل می‌بینیم اثر اغتشاش تنها در لحظه اعمال دیده می‌شود و در پاسخ حالت ماندگار سیستم تاثیری نمی‌گذارد و همچنان حتی با وجود اغتشاش خطای حالت ماندگار صفر است.

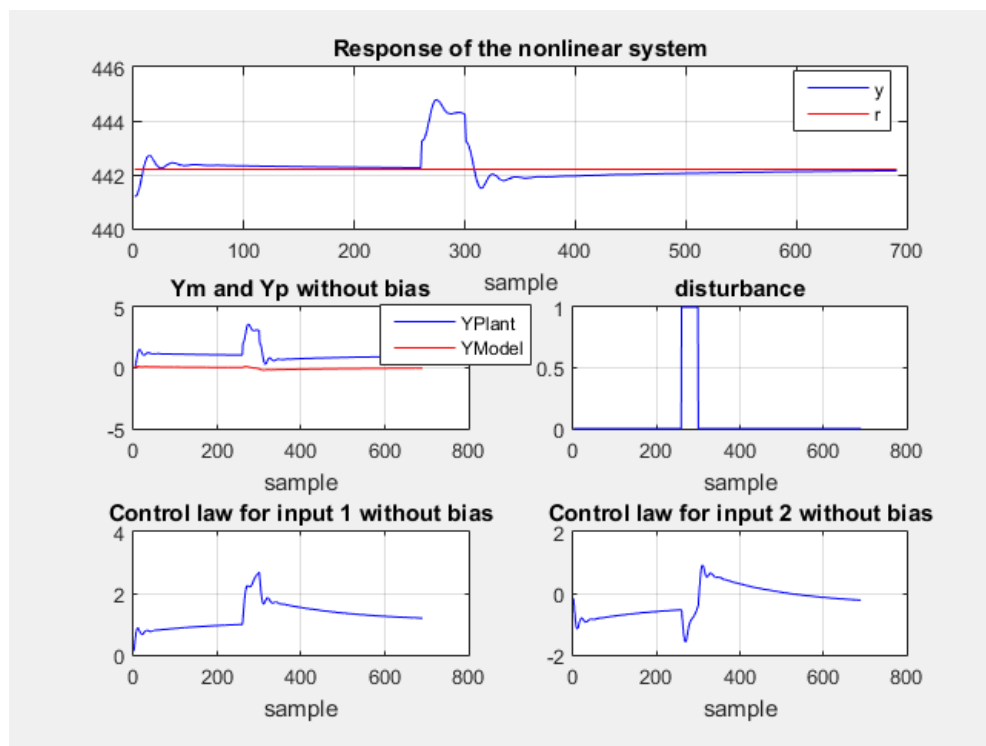


Figure 13. Responses of the system in verifying the effect of disturbance.

## 5.2. اثر نویز:

در این قسمت به خروجی پروسه نویز سفید با توان 0.01 اضافه می‌کنیم. پاسخ سیستم در حضور نویز سفید در زیر آمده است.

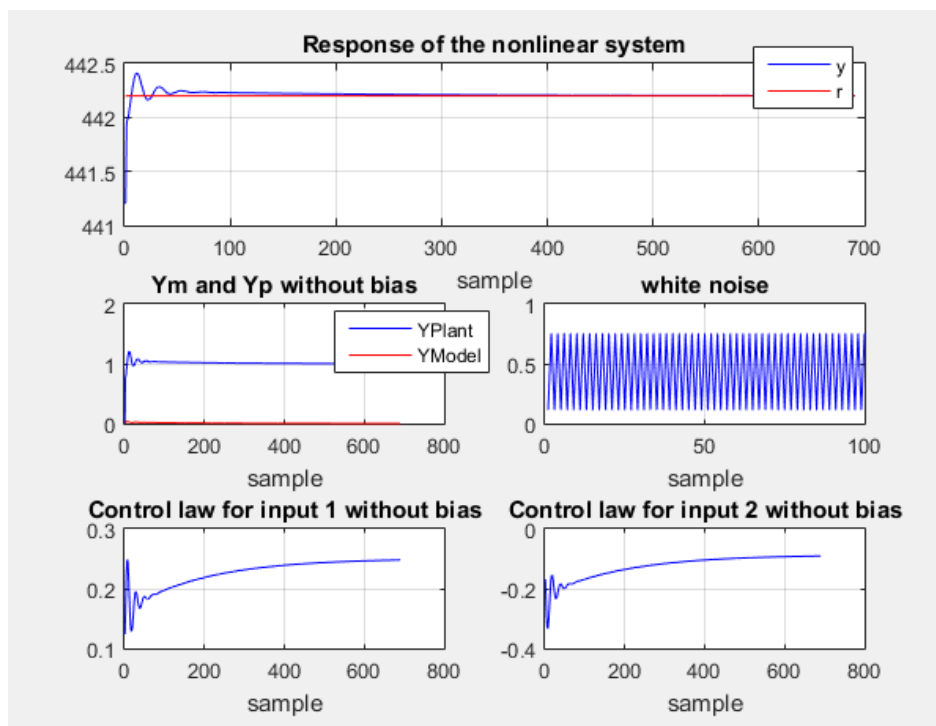


Figure 14. Responses of the system in verifying the effect of noise.

همانطور که در شکل بالا می‌بینیم سفید تنها پاسخ حالت گذرا را تحت تاثیر قرار داده و بر روی پاسخ حالت ماندگار تاثیری نگذاشته و خطای حالت ماندگار سیستم حتی با وجود نویز هم به صفر رسیده است.

## 6. بررسی عدم قطعیت در پروسه:

در این بخش به بررسی عدم قطعیت بر روی پاسخ سیستم می‌پردازیم. ابتدا به پارامتر  $V$  سیستم غیر خطی که همان پروسه می‌باشد عدم قطعیت با درصد های مختلف اضافه می‌کنیم. برای این پارامتر عدم قطعیت با درصد های 5% و 20% اضافه می‌کنیم. پاسخ سیستم به ازای این عدم قطعیت ها برای این پارامتر در شکل زیر آورده شده است.

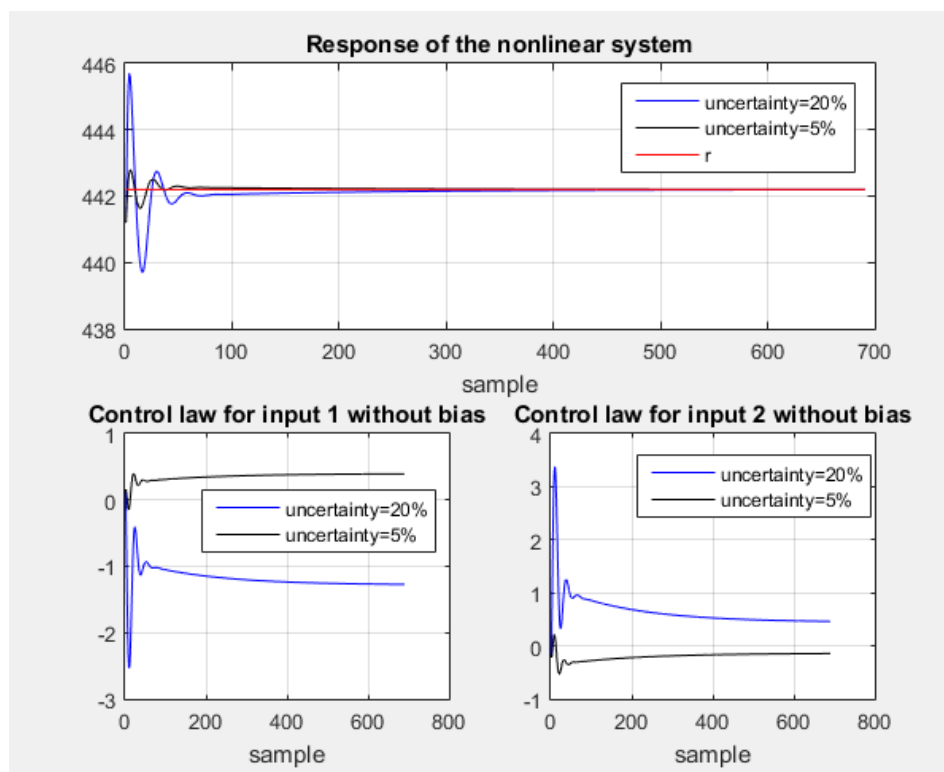


Figure 15. Responses of the system in verifying the effect of uncertainty on  $V$ .

همانطور که در شکل بالا می‌بینیم وجود عدم قطعیت تنها پاسخ حالت گذرا را تا حدودی تحت تاثیر قرار داده است، اما خطای حالت ماندگار را به صفر رسانده است. که این برتری PFC را نسبت به دو کنترل کننده MAC و DMC را نشان می‌دهد زیرا در این دو کنترل کننده عدم قطعیت بر روی پاسخ حالت ماندگار نیز تاثیر می‌گذاشت و بنابراین می‌بینیم که بازه تاثیر عدم قطعیت بر روی پاسخ سیستم بسیار بزرگ است تا جایی که حتی به ازای عدم قطعیت 20% پاسخ سیستم خطای حالت ماندگارش به صفر رسیده است.

اینبار عدم قطعیت با درصد های 5% و 20% را به پارامتر  $C_{A0}$  اضافه می‌کنیم. شکل زیر پاسخ سیستم در حضور عدم قطعیت های مذکور را نشان می‌دهد.

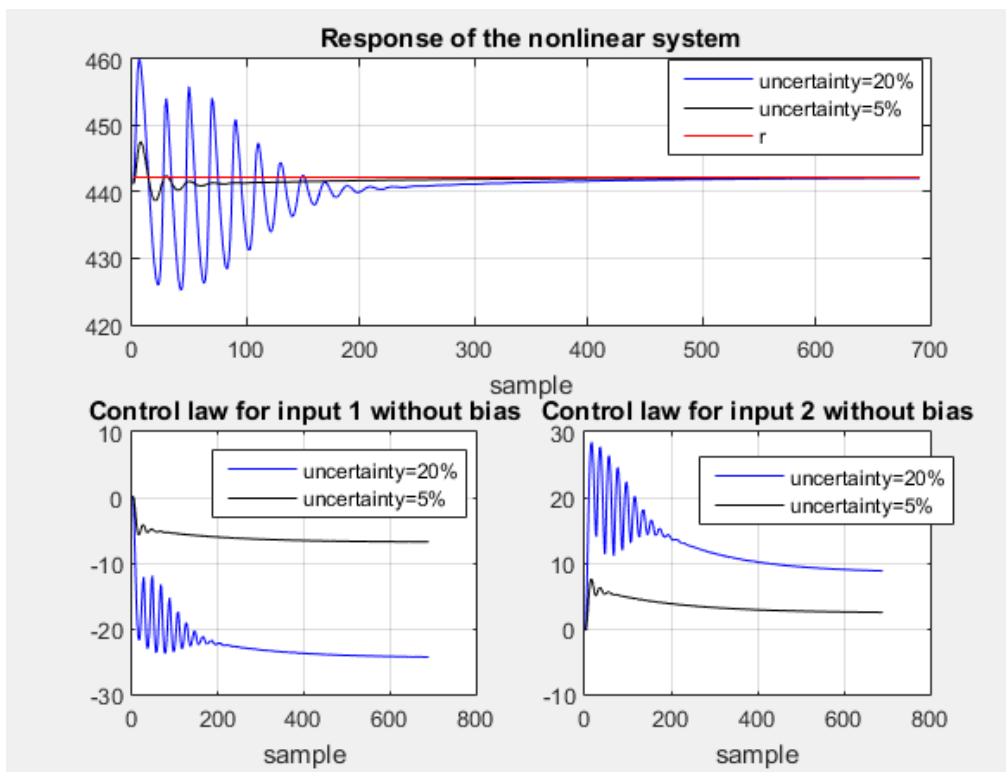


Figure 16. Responses of the system in verifying the effect of uncertainty on  $C_{A0}$ .

همانطور که در شکل بالا می‌بینیم وجود عدم قطعیت برای این پارامتر نیز تنها پاسخ حالت گذرا را تا حدودی تحت تاثیر قرار داده است، اما خطای حالت ماندگار را به صفر رسیده است. می‌بینیم که با افزایش درصد عدم قطعیت پاسخ حالت گذرا نوسانی‌تر شده است.

## 7. کدها و شبیه‌سازی‌ها:

کدها و شبیه‌سازی‌های مورد استفاده در این پروژه در زیر آورده شده است. کد زیر یک فانکشن است که به منظور فراخوانی سیستم خطی سازی شده نوشته شده است.

```
function [ a1,b1,a2,b2] = Inputsys(I)
if I==1
q=100; V=100; Cas=.0882; dH=2e5; ro=1e3; Cp=1; roc=1e3; Cpc=1; qc=100; ha=7e5; Ts=441.2; K0=7.2e10;
J=1e4; Ks=K0*exp(-J/Ts); Ca0=1; T0=350; Tc0=350; Ks_=K0*(exp(-J/Ts))*(J/(Ts^2));
a11=-q/V-Ks;
a12=-Cas*Ks_;
a21=-(-dH/(ro*Cp))*Ks;
a22=-q/V+(dH*Cas/(ro*Cp))*Ks_+(-roc*Cpc/(ro*Cp*V))*qc+(roc*Cpc/(ro*Cp*V))*qc*exp(-ha/(qc*ro*Cp));
b11=(Ca0-Cas)/V;
b12=0;
b21=(T0-Ts)/V;
b22=((roc*Cpc)/(ro*Cp*V))*(Tc0-Ts)*(qc*(-exp(-ha/(qc*roc*Cpc))*(ha/((qc^2)*roc*Cpc)))+(1-exp(-
ha/(qc*roc*Cpc))));
A=[a11 a12; a21 a22];
```

```

B=[b11 b12; b21 b22];
C=[0 1];
D=[0 0];
[a1,b1]=ss2tf(A,B,C,D,1);
[a2,b2]=ss2tf(A,B,C,D,2);
end
if I==2
    a1=1; b1=[1 1];
    a2=1; b2=[1 1];
end
if I==3
    a1=1; b1=[1 1];
    a2=0; b2=[1 1];
end
if I==4
q=1; V=1; Cas=0.265; dH=130*10e6; ro=1e6; Cp=1; roc=1e6; Cpc=1; qc=15; ha=7e5; Ts=393.954; K0=10e10;
J=8330; Ks=K0*exp(-J/Ts); Ca0=2; T0=350; Tc0=350; Ks_=K0*(exp(-J/Ts))*(J/(Ts^2));
a11=-q/V-Ks;
a12=-Cas*Ks_;
a21=-(dH/(ro*Cp))*Ks;
a22=-q/V+(-dH*Cas/(ro*Cp))*Ks_+(-roc*Cpc/(ro*Cp*V))*qc+(roc*Cpc/(ro*Cp*V))*qc*exp(-ha/(qc*ro*Cp));
b11=(Ca0-Cas)/V;
b12=0;
b21=(T0-Ts)/V;
b22=((roc*Cpc)/(ro*Cp*V))*(Tc0-Ts)*(qc*(-exp(-ha/(qc*roc*Cpc)))*(ha/((qc^2)*roc*Cpc)))+(1-exp(-
ha/(qc*roc*Cpc)));
A=[a11 a12; a21 a22];
B=[b11 b12; b21 b22];
C=[0 1];
D=[0 0];
[a1,b1]=ss2tf(A,B,C,D,1);
[a2,b2]=ss2tf(A,B,C,D,2);
end
end

```

کدهای زیر مربوط به قسمتی است که خروجی مطلوب های مختلف به سیستم داده شده است. همچنین بررسی عدم قطعیت، نویز و اغتشاش توسط این کد انجام شده است. لازم به ذکر است که برای انجام هر قسمت بایستی کد مربوط به آن قسمت `uncomment` شود و بعد مورد استفاده قرار گیرد.

```

clear
clc
[n1,d1,n2,d2]=Inputsys(1);
Gs1 = tf(n1,d1);
Ts=0.1;
Gd1 = c2d(Gs1,Ts,'zoh');
[num1,den1]=tfdata(Gd1,'v');
Gs2 = tf(n2,d2);
Gd2 = c2d(Gs2,Ts,'zoh');
[num2,den2]=tfdata(Gd2,'v');
sys_info = stepinfo(Gd1);
ts1 = sys_info.SettlingTime;
tr1=sys_info.RiseTime;
sys_info = stepinfo(Gd2);
ts2 = sys_info.SettlingTime;
tr2=sys_info.RiseTime;
t=1:Ts:80;

```

```

[g1,t1] = step(Gd1,t);
[g2,t2] = step(Gd2,t);
P1=floor(tr1/Ts);
P2=floor(tr2/Ts);
N1=floor( ts1/Ts);
N2=floor( ts2/Ts);
P=max(P1,P2);
N=max(N1,N2);
M=P;
n=3;
m1=2;
miu=[3 5];
ui1=randn(1,length(t));
ui2=randn(1,length(t));
ui=[ui1; ui2];
Gd=[Gd1 Gd2];
[yi,t1]=lsim(Gd,ui,t);
theta1=zeros(2*miu(1)+3*n,length(t)-100);
theta2=zeros(2*miu(2)+3*n,length(t)-100);
for i=1:2
    phi1=zeros(2*miu(i)+3*n,length(t));
    for j=1:length(t)-100
        y(1,j)=yi(j+miu(i));
        for k=1:miu(i)
            if (j+miu(i)-k-1)<=0
                du(j+miu(i)-k)=ui1(j+miu(i)-k);
                du2(j+miu(i)-k)=ui2(j+miu(i)-k);
            else
                du(j+miu(i)-k)=ui1(j+miu(i)-k)-ui1(j+miu(i)-k-1);
                du2(j+miu(i)-k)=ui2(j+miu(i)-k)-ui2(j+miu(i)-k-1);
            end
            phi1(k,j)=du(j+miu(i)-k);
            phi1(k+miu(i),j)=du2(j+miu(i)-k);
        end
        for k=1:n
            if (j-k-1)<=0
                phi1(k+2*miu(i),j)=0;
                phi1(k+2*miu(i)+n,j)=0;
            else
                phi1(k+2*miu(i),j)=ui1(j-k)-ui1(j-k-1);
                phi1(k+2*miu(i)+n,j)=ui2(j-k)-ui2(j-k-1);
            end
            if (j-k)<=0
                phi1(k+2*miu(i)+2*n,j)=0;
            else
                phi1(k+2*miu(i)+2*n,j)=yi(j-k);
            end
        end
        if i==1
            %theta1(:,j)=(phi1(:,j)*phi1(:,j))\'\'(phi1(:,j)*y(1,j)); % we could use this method (right inverse)
            theta1(:,j)=pinv(phi1(:,j))\'\'*y(1,j);
        else
            %theta2(:,j)=(phi1(:,j)*phi1(:,j))\'\'(phi1(:,j)*y(1,j)); % we could use this method (right inverse)
            theta2(:,j)=pinv(phi1(:,j))\'\'*y(1,j);
        end
    end
end

```



```

end
% .....
G1i=zeros(2,2,length(t)-100); % these comments are related to the state which we use IIR for the future
G2i=zeros(2,2,length(t)-100);
M1_=zeros(2,n,length(t)-100);
M2_=zeros(2,n,length(t)-100);
F=zeros(2,n,length(t)-100);
for q=1:length(t)-100
    G1i(:,q)=[theta1(miu(1),q) theta1(miu(1)-m1,q); theta2(miu(2),q) theta2(miu(2)-m1,q)];
    G2i(:,q)=[theta1(2*miu(1),q) theta1(2*miu(1)-m1,q); theta2(2*miu(2),q) theta2(2*miu(2)-m1,q)];
    M1_(:,q)=[theta1(2*miu(1)+1:n+2*miu(1),q)'; theta2(2*miu(2)+1:n+2*miu(2),q)'];
    M2_(:,q)=[theta1(2*miu(1)+1+n:2*n+2*miu(1),q)'; theta2(2*miu(2)+n+1:2*n+2*miu(2),q)'];
    F(:,q)=[theta1(2*miu(1)+2*n+1:3*n+2*miu(1),q)'; theta2(2*miu(2)+2*n+1:2*miu(2)+3*n,q)'];
end
Gi=[G1i G2i];
% .....Toeplitz Matrix.....
b1 = zeros(1,P); b1(1,1)= g1(2);
a1 = g1(2:P+1);
G1 = toeplitz(a1,b1);
G1(:,M) = G1(:,M:P)*ones(P-M+1,1);
G1 = G1(:,1:M);
% .....
b2 = zeros(1,P); b2(1,1)= g2(2);
a2 = g2(2:P+1);
G2 = toeplitz(a2,b2);
G2(:,M) = G2(:,M:P)*ones(P-M+1,1);
G2 = G2(:,1:M);
% .....
G11=[G1(miu(1),1) G1(miu(1),m1+1); G1(miu(2),1) G1(miu(2),m1+1)];
G21=[G2(miu(1),1) G2(miu(1),m1+1); G2(miu(2),1) G2(miu(2),m1+1)];
G=[G11 G21];
M_=[M1_ M2_];
% .....
gamma =1/60;
gain_DC=(num1(1)+num1(2)+num1(3))/(den1(1)+den1(2)+den1(3));
gain_DC2=(num2(1)+num2(2)+num2(3))/(den2(1)+den2(2)+den2(3));
Q = eye(2);
R1 =((1.2)^2)*gamma*gain_DC^2*eye(2);
R2=gamma*gain_DC2^2*eye(2);
R=[R1 zeros(2); zeros(2) R2];
alpha=0.5;
%for q=1:length(t)-100
    Kpfc=(G'*Q*G+R)\(G'*Q);
%end
% .....

x01=0.0882;
x02=441.2;
dU1_=zeros(n,length(t)-100);
dU2_=zeros(n,length(t)-100);
dU_=[dU1_;dU2_];
d1=zeros(1,length(t)-100);
y1=441.2;
u_1=[];
u_2=[];
ym=[];

```

17

18

```

E(:,i+1)=Y_d(:,i+1)-Y_past(:,i+1)-D(:,i+1);

dU(:,i+1)=Kpfc*E(:,i+1);
dU1(:,i+1)=dU(1:2,i+1);
dU2(:,i+1)=dU(3:4,i+1);
U1(1,i+1)=dU1(1,i+1)+U1(1,i);
U2(1,i+1)=dU2(1,i+1)+U2(1,i);
dU(:,i+1)=[dU1(:,i+1);dU2(:,i+1)];

Y_m(:,i+1)=G*dU(:,i+1)+Y_past(:,i+1);

dU1_(2:n,i+2) = dU1_(1:n-1,i+1);
dU1_(1,i+2)=dU1(1,i+1);
dU2_(2:n,i+2) = dU2_(1:n-1,i+1);
dU2_(1,i+2)=dU2(1,i+1);
dU_(:,i+2)=[dU1_(:,i+2);dU2_(:,i+2)];
%Y_(2:n,i+2+miu(1))=Y_(1:n-1,i+1+miu(1)); % without using y(t+1) we can make Y_past in this way
% Y_(1,i+2+miu(1))=Y_m(1,i+1);

u1=U1(1,i+1);
u2=U2(1,i+1);
sim('Model');
d1(i+2)=y(end)-yl(end);
Y_(2:n,i+2)=Y_(1:n-1,i+1);
Y_(1,i+2)=yl(end);
y=y(end);%+dist(i,1); % nonlinear
y1=[y1; y+441.2];
ym=[ym; yl(end)];
u_1=[u_1; u1];
u_2=[u_2; u2];
%noise=[noise; n];
x01=x1(end);
x02=x2(end);

end

figure(1);
subplot(2,2,1);
plot(y1,'b');
hold on
plot(r+441.2,'r');
grid on
%axis([0 2200 435 450]);
legend('y','r');
title('Response of the nonlinear system');
xlabel('sample');
%figure(4);
subplot(2,2,2);
plot(y1-441.2,'b');
hold on
plot(ym,'r');
grid on
xlabel('sample');
title('Ym and Yp without bias');
legend('YPlant','YModel');

```

```

%figure(5);
subplot(2,2,3);
plot(u_1,'b');
grid on
xlabel('sample');
title('Control law for input 1 without bias');
figure(6);
subplot(2,2,4);
plot(u_2,'b');
grid on
xlabel('sample');
title('Control law for input 2 without bias');
%subplot(3,2,4);
% plot(dist,'b');
% grid on
% xlabel('sample');
% title('disturbance');
% plot(noise,'b');
% grid on
% xlabel('sample');
% title('white noise');
% axis([0 100 0 1]);

```

شبیه سازی زیر مربوط به قسمتی است که می خواهیم خروجی پروسه را فراخوانی کنیم و  $u$  های بدست آمده از کنترلر به این پروسه با یک بایاسی داده می شوند. هنگامی که می خواهیم اثر نویز سفید را بررسی کنیم بایستی بلوک نویز سفید را به جمع کننده متصل نماییم.

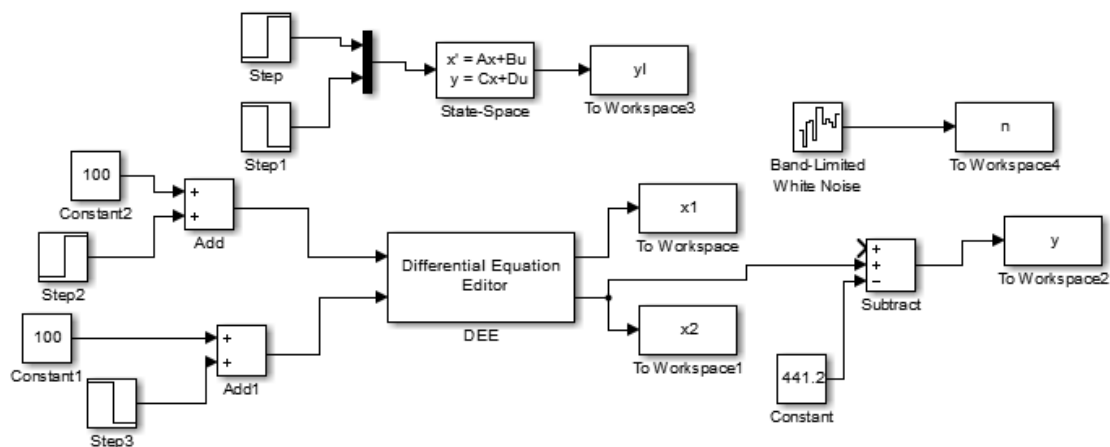


Figure 17. Simulation of the linear and nonlinear plant in Simulink.

کدهای زیر برای بررسی پارامترهای مختلف سیستم نوشته شده اند.

```

% varying gamma.....
% .....
clear
clc
[n1,d1,n2,d2]=Inputs(1);
Gs1 = tf(n1,d1);
Ts=0.1;
Gd1 = c2d(Gs1,Ts,'zoh');
[num1,den1]=tfdata(Gd1,'v');

```

```

Gs2 = tf(n2,d2);
Gd2 = c2d(Gs2,Ts,'zoh');
[num2,den2]=tfdata(Gd2,'v');
sys_info = stepinfo(Gd1);
ts1 = sys_info.SettlingTime;
tr1=sys_info.RiseTime;
sys_info = stepinfo(Gd2);
ts2 = sys_info.SettlingTime;
tr2=sys_info.RiseTime;
t=1:Ts:80;
[g1,t1] = step(Gd1,t);
[g2,t2] = step(Gd2,t);
P1=floor(tr1/Ts);
P2=floor(tr2/Ts);
N1=floor( ts1/Ts);
N2=floor( ts2/Ts);
P=max(P1,P2);
N=max(N1,N2);
M=P;
n=3;
m1=2;
miu=[3 5];
ui1=randn(1,length(t));
ui2=randn(1,length(t));
ui=[ui1; ui2];
Gd=[Gd1 Gd2];
[yi,t1]=lsim(Gd,ui,t);
theta1=zeros(2*miu(1)+3*n,length(t)-100);
theta2=zeros(2*miu(2)+3*n,length(t)-100);
for i=1:2
    phi1=zeros(2*miu(i)+3*n,length(t));
    for j=1:length(t)-100
        y(1,j)=yi(j+miu(i));
        for k=1:miu(i)
            if (j+miu(i)-k-1)<=0
                du(j+miu(i)-k)=ui1(j+miu(i)-k);
                du2(j+miu(i)-k)=ui2(j+miu(i)-k);
            else
                du(j+miu(i)-k)=ui1(j+miu(i)-k)-ui1(j+miu(i)-k-1);
                du2(j+miu(i)-k)=ui2(j+miu(i)-k)-ui2(j+miu(i)-k-1);
            end
            phi1(k,j)=du(j+miu(i)-k);
            phi1(k+miu(i),j)=du2(j+miu(i)-k);
        end
        for k=1:n
            if (j-k-1)<=0
                phi1(k+2*miu(i),j)=0;
                phi1(k+2*miu(i)+n,j)=0;
            else
                phi1(k+2*miu(i),j)=ui1(j-k)-ui1(j-k-1);
                phi1(k+2*miu(i)+n,j)=ui2(j-k)-ui2(j-k-1);
            end
            if (j-k)<=0
                phi1(k+2*miu(i)+2*n,j)=0;
            else
                phi1(k+2*miu(i)+2*n,j)=yi(j-k);
            end
        end
    end
end

```

```

        end
    end
    if i==1
        %theta1(:,j)=(phi1(:,j)*phi1(:,j'))/(phi1(:,j)*y(1,j)); % we could use this method (right inverse)
        theta1(:,j)=pinv(phi1(:,j))*y(1,j);
    else
        %theta2(:,j)=(phi1(:,j)*phi1(:,j'))/(phi1(:,j)*y(1,j)); % we could use this method (right inverse)
        theta2(:,j)=pinv(phi1(:,j))*y(1,j);
    end
end
end
%.....
G1i=zeros(2,2,length(t)-100); % these comments are related to the state which we use IIR for the future
G2i=zeros(2,2,length(t)-100);
M1_=zeros(2,n,length(t)-100);
M2_=zeros(2,n,length(t)-100);
F=zeros(2,n,length(t)-100);
for q=1:length(t)-100
    G1i(:,q)=[theta1(miu(1),q) theta1(miu(1)-m1,q); theta2(miu(2),q) theta2(miu(2)-m1,q)];
    G2i(:,q)=[theta1(2*miu(1),q) theta1(2*miu(1)-m1,q); theta2(2*miu(2),q) theta2(2*miu(2)-m1,q)];
    M1_(:,q)=[theta1(2*miu(1)+1:n+2*miu(1),q)'; theta2(2*miu(2)+1:n+2*miu(2),q)'];
    M2_(:,q)=[theta1(2*miu(1)+1:n+2*miu(1),q)'; theta2(2*miu(2)+1:n+2*miu(2),q)'];
    F(:,q)=[theta1(2*miu(1)+2*n+1:3*n+2*miu(1),q)'; theta2(2*miu(2)+2*n+1:2*miu(2)+3*n,q)'];
end
Gi=[G1i G2i];
%.....Toeplitz Matrix.....
b1 = zeros(1,P); b1(1,1)= g1(2);
a1 = g1(2:P+1);
G1 = toeplitz(a1,b1);
G1(:,M) = G1(:,M:P)*ones(P-M+1,1);
G1 = G1(:,1:M);
%.....
b2 = zeros(1,P); b2(1,1)= g2(2);
a2 = g2(2:P+1);
G2 = toeplitz(a2,b2);
G2(:,M) = G2(:,M:P)*ones(P-M+1,1);
G2 = G2(:,1:M);
%.....
G11=[G1(miu(1),1) G1(miu(1),m1+1); G1(miu(2),1) G1(miu(2),m1+1)];
G21=[G2(miu(1),1) G2(miu(1),m1+1); G2(miu(2),1) G2(miu(2),m1+1)];
G=[G11 G21];
M_=[M1_ M2_];
%.....
%..... gamma=1.....
gamma =1;
gain_DC=(num1(1)+num1(2)+num1(3))/(den1(1)+den1(2)+den1(3));
gain_DC2=(num2(1)+num2(2)+num2(3))/(den2(1)+den2(2)+den2(3));
Q = eye(2);
R1=((1.2)^2)*gamma*gain_DC^2*eye(2);
R2=gamma*gain_DC2^2*eye(2);
R=[R1 zeros(2); zeros(2) R2];
alpha=0.5;
%for q=1:length(t)-100
    Kpfc=(G'*Q*G+R)\(G'*Q);
%end
%.....

```

```

x01=0.0882;
x02=441.2;
dU1_=zeros(n,length(t)-100);
dU2_=zeros(n,length(t)-100);
dU_=[dU1_;dU2_];
d1=zeros(1,length(t)-100);
y1=441.2;
u_1=[];
u_2=[];
ym=[];
y=0;
Y_d=zeros(2,length(t)-100);
Y_past=zeros(2,length(t)-100);
Y_m=zeros(2,length(t)-100);
D=zeros(2,length(t)-100);
E=zeros(2,length(t)-100);
dU1=zeros(2,length(t)-100);
dU2=zeros(2,length(t)-100);
dU=[dU1;dU2];
U1=zeros(M,length(t)-100);
U2=zeros(M,length(t)-100);
Y_=zeros(n,length(t)-100);
% .....step.....
r =ones(length(t)-100,1);
% .....
for i=1:length(t)-101

for j=1:2
    Y_d(j,i+1)=(alpha^j)*y+(1-(alpha)^j)*r(i+1); % Programmed
end

Y_past(:,i+1)=M_(:,i+1)*dU_(:,i+1)+F(:,i+1)*Y_(:,i+1);
D(:,i+1)=d1(i+1)*ones(2,1);

E(:,i+1)=Y_d(:,i+1)-Y_past(:,i+1)-D(:,i+1);

dU(:,i+1)=Kpfc*E(:,i+1);
dU1(:,i+1)=dU(1:2,i+1);
dU2(:,i+1)=dU(3:4,i+1);
U1(1,i+1)=dU1(1,i+1)+U1(1,i);
U2(1,i+1)=dU2(1,i+1)+U2(1,i);
dU(:,i+1)=[dU1(:,i+1);dU2(:,i+1)];

Y_m(:,i+1)=G*dU(:,i+1)+Y_past(:,i+1);

dU1_(2:n,i+2) = dU1_(1:n-1,i+1);
dU1_(1,i+2)=dU1(1,i+1);
dU2_(2:n,i+2) = dU2_(1:n-1,i+1);
dU2_(1,i+2)=dU2(1,i+1);
dU_(:,i+2)=[dU1_(:,i+2);dU2_(:,i+2)];
% Y_(2:n,i+2+miu(1))=Y_(1:n-1,i+1+miu(1)); % without using y(t+1) we can make Y_past in this way
% Y_(1,i+2+miu(1))=Y_m(1,i+1);

```



```

u1=U1(1,i+1);
u2=U2(1,i+1);
sim('Model');
d1(i+2)=y(end)-yl(end);
Y_(2:n,i+2)=Y_(1:n-1,i+1);
Y_(1,i+2)=yl(end);
y=y(end);%+dist(i,1); % nonlinear
y1=[y1; y+441.2];
ym=[ym; yl(end)];
u_1=[u_1; u1];
u_2=[u_2; u2];
%noise=[noise; n];
x01=x1(end);
x02=x2(end);

end
figure(3);
subplot(2,2,1:2);
plot(y1,'b');
grid on
title('Response of the nonlinear system');
xlabel('sample');
subplot(2,2,3);
plot(u_1,'b');
grid on
xlabel('sample');
title('Control law for input 1 without bias');
subplot(2,2,4);
plot(u_2,'b');
grid on
xlabel('sample');
title('Control law for input 2 without bias');
% .....
% ..... gamma=1/60 .....
gamma =1/60;
gain_DC=(num1(1)+num1(2)+num1(3))/(den1(1)+den1(2)+den1(3));
gain_DC2=(num2(1)+num2(2)+num2(3))/(den2(1)+den2(2)+den2(3));
Q = eye(2);
R1=((1.2)^2)*gamma*gain_DC^2*eye(2);
R2=gamma*gain_DC2^2*eye(2);
R=[R1 zeros(2); zeros(2) R2];
alpha=0.5;
%for q=1:length(t)-100
    Kpfc=(G'*Q*G+R)\(G'*Q);
%end
% .....
n=3;
x01=0.0882;
x02=441.2;
dU1_=zeros(n,length(t)-100);
dU2_=zeros(n,length(t)-100);
dU_=[dU1_;dU2_];
d1=zeros(1,length(t)-100);
y1=441.2;
u_1=[];
u_2=[];

```

```

ym=[];
y=0;
Y_d=zeros(2,length(t)-100);
Y_past=zeros(2,length(t)-100);
Y_m=zeros(2,length(t)-100);
D=zeros(2,length(t)-100);
E=zeros(2,length(t)-100);
dU1=zeros(2,length(t)-100);
dU2=zeros(2,length(t)-100);
dU=[dU1;dU2];
U1=zeros(M,length(t)-100);
U2=zeros(M,length(t)-100);
Y_=zeros(n,length(t)-100);
%.....step.....
r =ones(length(t)-100,1);
%.....
for i=1:length(t)-101

for j=1:2
    Y_d(j,i+1)=(alpha^j)*y+(1-(alpha)^j)*r(i+1); % Programmed
end

Y_past(:,i+1)=M(:,i+1)*dU(:,i+1)+F(:,i+1)*Y_(:,i+1);
D(:,i+1)=d1(i+1)*ones(2,1);

E(:,i+1)=Y_d(:,i+1)-Y_past(:,i+1)-D(:,i+1);

dU(:,i+1)=Kpfc*E(:,i+1);
dU1(:,i+1)=dU(1:2,i+1);
dU2(:,i+1)=dU(3:4,i+1);
U1(1,i+1)=dU1(1,i+1)+U1(1,i);
U2(1,i+1)=dU2(1,i+1)+U2(1,i);
dU(:,i+1)=[dU1(:,i+1);dU2(:,i+1)];

Y_m(:,i+1)=G*dU(:,i+1)+Y_past(:,i+1);

dU1_(2:n,i+2) = dU1_(1:n-1,i+1);
dU1_(1,i+2)=dU1(1,i+1);
dU2_(2:n,i+2) = dU2_(1:n-1,i+1);
dU2_(1,i+2)=dU2(1,i+1);
dU_(:,i+2)=[dU1_(:,i+2);dU2_(:,i+2)];
% Y_(2:n,i+2+miu(1))=Y_(1:n-1,i+1+miu(1)); % without using y(t+1) we can make Y_past in this way
% Y_(1,i+2+miu(1))=Y_m(1,i+1);

u1=U1(1,i+1);
u2=U2(1,i+1);
sim('Model');
d1(i+2)=y(end)-yl(end);
Y_(2:n,i+2)=Y_(1:n-1,i+1);
Y_(1,i+2)=yl(end);
y=y(end);%+dist(i,1); % nonlinear
yl=[y1; y+441.2];
ym=[ym; yl(end)];
u_1=[u_1; u1];

```

```

u_2=[u_2; u2];
%noise=[noise; n];
x01=x1(end);
x02=x2(end);

end
figure(3);
subplot(2,2,1:2);
hold on
plot(y1,'c');
subplot(2,2,3);
hold on
plot(u_1,'c');
subplot(2,2,4);
hold on
plot(u_2,'c');
% .....
% ..... gamma=0.006 .....
gamma =0.006;
gain_DC=(num1(1)+num1(2)+num1(3))/(den1(1)+den1(2)+den1(3));
gain_DC2=(num2(1)+num2(2)+num2(3))/(den2(1)+den2(2)+den2(3));
Q = eye(2);
R1 =((1.2)^2)*gamma*gain_DC^2*eye(2);
R2=gamma*gain_DC2^2*eye(2);
R=[R1 zeros(2); zeros(2) R2];
alpha=0.5;
%for q=1:length(t)-100
    Kpfc=(G'*Q*G+R)\(G'*Q);
%end
% .....
n=3;
x01=0.0882;
x02=441.2;
dU1_=zeros(n,length(t)-100);
dU2_=zeros(n,length(t)-100);
dU_=[dU1_;dU2_];
d1=zeros(1,length(t)-100);
y1=441.2;
u_1=[];
u_2=[];
ym=[];
y=0;
Y_d=zeros(2,length(t)-100);
Y_past=zeros(2,length(t)-100);
Y_m=zeros(2,length(t)-100);
D=zeros(2,length(t)-100);
E=zeros(2,length(t)-100);
dU1=zeros(2,length(t)-100);
dU2=zeros(2,length(t)-100);
dU=[dU1;dU2];
U1=zeros(M,length(t)-100);
U2=zeros(M,length(t)-100);
Y_=zeros(n,length(t)-100);
% .....step .....
r =ones(length(t)-100,1);
% .....

```

```

for i=1:length(t)-101

for j=1:2
    Y_d(j,i+1)=(alpha^j)*y+(1-(alpha)^j)*r(i+1); % Programmed
end

Y_past(:,i+1)=M_(:,i+1)*dU_(:,i+1)+F(:,i+1)*Y_(:,i+1);
D(:,i+1)=d1(i+1)*ones(2,1);

E(:,i+1)=Y_d(:,i+1)-Y_past(:,i+1)-D(:,i+1);

dU(:,i+1)=Kpfc*E(:,i+1);
dU1(:,i+1)=dU(1:2,i+1);
dU2(:,i+1)=dU(3:4,i+1);
U1(1,i+1)=dU1(1,i+1)+U1(1,i);
U2(1,i+1)=dU2(1,i+1)+U2(1,i);
dU(:,i+1)=[dU1(:,i+1);dU2(:,i+1)];

Y_m(:,i+1)=G*dU(:,i+1)+Y_past(:,i+1);

dU1_(2:n,i+2) = dU1_(1:n-1,i+1);
dU1_(1,i+2)=dU1(1,i+1);
dU2_(2:n,i+2) = dU2_(1:n-1,i+1);
dU2_(1,i+2)=dU2(1,i+1);
dU_(:,i+2)=[dU1_(:,i+2);dU2_(:,i+2)];
% Y_(2:n,i+2+miu(1))=Y_(1:n-1,i+1+miu(1)); % without using y(t+1) we can make Y_past in this way
% Y_(1,i+2+miu(1))=Y_m(1,i+1);

u1=U1(1,i+1);
u2=U2(1,i+1);
sim('Model');
d1(i+2)=y(end)-yl(end);
Y_(2:n,i+2)=Y_(1:n-1,i+1);
Y_(1,i+2)=yl(end);
y=y(end);%+dist(i,1); % nonlinear
y1=[y1; y+441.2];
ym=[ym; yl(end)];
u_1=[u_1; u1];
u_2=[u_2; u2];
%noise=[noise; n];
x01=x1(end);
x02=x2(end);

end
figure(3);
subplot(2,2,1:2);
hold on
plot(y1,'m');
hold on
plot(r+441.2,'r');
legend('gamma=1','gamma=1/60','gamma=0.006','r');

subplot(2,2,3);
hold on

```

```

plot(u_1,'m');
legend('gamma=1','gamma=1/60','gamma=0.006');
subplot(2,2,4);
hold on
plot(u_2,'m');
legend('gamma=1','gamma=1/60','gamma=0.006');

% veriying Q.....
%.....
clear
clc
[n1,d1,n2,d2]=Inputsys(1);
Gs1 = tf(n1,d1);
Ts=0.1;
Gd1 = c2d(Gs1,Ts,'zoh');
[num1,den1]=tfdata(Gd1,'v');
Gs2 = tf(n2,d2);
Gd2 = c2d(Gs2,Ts,'zoh');
[num2,den2]=tfdata(Gd2,'v');
sys_info = stepinfo(Gd1);
ts1 = sys_info.SettlingTime;
tr1=sys_info.RiseTime;
sys_info = stepinfo(Gd2);
ts2 = sys_info.SettlingTime;
tr2=sys_info.RiseTime;
t=1:Ts:80;
[g1,t1] = step(Gd1,t);
[g2,t2] = step(Gd2,t);
P1=floor(tr1/Ts);
P2=floor(tr2/Ts);
N1=floor( ts1/Ts);
N2=floor( ts2/Ts);
P=max(P1,P2);
N=max(N1,N2);
M=P;
n=3;
m1=2;
miu=[3 5];
ui1=randn(1,length(t));
ui2=randn(1,length(t));
ui=[ui1; ui2];
Gd=[Gd1 Gd2];
[yi,t1]=lsim(Gd,ui,t);
theta1=zeros(2*miu(1)+3*n,length(t)-100);
theta2=zeros(2*miu(2)+3*n,length(t)-100);
for i=1:2
    phi1=zeros(2*miu(i)+3*n,length(t));
    for j=1:length(t)-100
        y(1,j)=yi(j+miu(i));
        for k=1:miu(i)
            if (j+miu(i)-k-1)<=0
                du(j+miu(i)-k)=ui1(j+miu(i)-k);
                du2(j+miu(i)-k)=ui2(j+miu(i)-k);
            else
                du(j+miu(i)-k)=ui1(j+miu(i)-k)-ui1(j+miu(i)-k-1);
                du2(j+miu(i)-k)=ui2(j+miu(i)-k)-ui2(j+miu(i)-k-1);
            end
        end
    end
end

```

```

    end
    phi1(k,j)=du(j+miu(i)-k);
    phi1(k+miu(i),j)=du2(j+miu(i)-k);
end
for k=1:n
    if (j-k-1)<=0
        phi1(k+2*miu(i),j)=0;
        phi1(k+2*miu(i)+n,j)=0;
    else
        phi1(k+2*miu(i),j)=ui1(j-k)-ui1(j-k-1);
        phi1(k+2*miu(i)+n,j)=ui2(j-k)-ui2(j-k-1);
    end
    if (j-k)<=0
        phi1(k+2*miu(i)+2*n,j)=0;
    else
        phi1(k+2*miu(i)+2*n,j)=yi(j-k);
    end
end
if i==1
    %theta1(:,j)=(phi1(:,j)*phi1(:,j'))\ (phi1(:,j)*y(1,j)); % we could use this method (right inverse)
    theta1(:,j)=pinv(phi1(:,j'))*y(1,j);
else
    %theta2(:,j)=(phi1(:,j)*phi1(:,j'))\ (phi1(:,j)*y(1,j)); % we could use this method (right inverse)
    theta2(:,j)=pinv(phi1(:,j'))*y(1,j);
end
end
end
%.....
G1i=zeros(2,2,length(t)-100); % these comments are related to the state which we use IIR for the future
G2i=zeros(2,2,length(t)-100);
M1_=zeros(2,n,length(t)-100);
M2_=zeros(2,n,length(t)-100);
F=zeros(2,n,length(t)-100);
for q=1:length(t)-100
    G1i(:,q)=[theta1(miu(1),q) theta1(miu(1)-m1,q); theta2(miu(2),q) theta2(miu(2)-m1,q)];
    G2i(:,q)=[theta1(2*miu(1),q) theta1(2*miu(1)-m1,q); theta2(2*miu(2),q) theta2(2*miu(2)-m1,q)];
    M1_(:,q)=[theta1(2*miu(1)+1:n+2*miu(1),q)'; theta2(2*miu(2)+1:n+2*miu(2),q)'];
    M2_(:,q)=[theta1(2*miu(1)+1+n:2*n+2*miu(1),q)'; theta2(2*miu(2)+n+1:2*n+2*miu(2),q)'];
    F(:,q)=[theta1(2*miu(1)+2*n+1:3*n+2*miu(1),q)'; theta2(2*miu(2)+2*n+1:2*miu(2)+3*n,q)'];
end
Gi=[G1i G2i];
%.....Toeplitz Matrix.....
b1 = zeros(1,P); b1(1,1)= g1(2);
a1 = g1(2:P+1);
G1 = toeplitz(a1,b1);
G1(:,M) = G1(:,M:P)*ones(P-M+1,1);
G1 = G1(:,1:M);
%.....
b2 = zeros(1,P); b2(1,1)= g2(2);
a2 = g2(2:P+1);
G2 = toeplitz(a2,b2);
G2(:,M) = G2(:,M:P)*ones(P-M+1,1);
G2 = G2(:,1:M);
%.....
G11=[G1(miu(1),1) G1(miu(1),m1+1); G1(miu(2),1) G1(miu(2),m1+1)];
G21=[G2(miu(1),1) G2(miu(1),m1+1); G2(miu(2),1) G2(miu(2),m1+1)];

```

```

G=[G11 G21];
M_=[M1_ M2_];
% .....
% ..... Q=I .....
gamma =1;
gain_DC=(num1(1)+num1(2)+num1(3))/(den1(1)+den1(2)+den1(3));
gain_DC2=(num2(1)+num2(2)+num2(3))/(den2(1)+den2(2)+den2(3));
Q = eye(2);
R1 =((1.2)^2)*gamma*gain_DC^2*eye(2);
R2=gamma*gain_DC2^2*eye(2);
R=[R1 zeros(2); zeros(2) R2];
alpha=0.5;
%for q=1:length(t)-100
    Kpfc=(G'*Q*G+R)\(G'*Q);
%end
% .....

```

```

x01=0.0882;
x02=441.2;
dU1_=zeros(n,length(t)-100);
dU2_=zeros(n,length(t)-100);
dU_=[dU1_;dU2_];
d1=zeros(1,length(t)-100);
y1=441.2;
u_1=[];
u_2=[];
ym=[];
y=0;
Y_d=zeros(2,length(t)-100);
Y_past=zeros(2,length(t)-100);
Y_m=zeros(2,length(t)-100);
D=zeros(2,length(t)-100);
E=zeros(2,length(t)-100);
dU1=zeros(2,length(t)-100);
dU2=zeros(2,length(t)-100);
dU=[dU1;dU2];
U1=zeros(M,length(t)-100);
U2=zeros(M,length(t)-100);
Y_=zeros(n,length(t)-100);
% .....step .....
r =ones(length(t)-100,1);
% .....
for i=1:length(t)-101

for j=1:2
    Y_d(j,i+1)=(alpha^j)*y+(1-(alpha)^j)*r(i+1); % Programmed
end

Y_past(:,i+1)=M_(:,i+1)*dU_(:,i+1)+F(:,i+1)*Y_(:,i+1);
D(:,i+1)=d1(i+1)*ones(2,1);

E(:,i+1)=Y_d(:,i+1)-Y_past(:,i+1)-D(:,i+1);

dU(:,i+1)=Kpfc*E(:,i+1);
dU1(:,i+1)=dU(1:2,i+1);

```

```

dU2(:,i+1)=dU(3:4,i+1);
U1(1,i+1)=dU1(1,i+1)+U1(1,i);
U2(1,i+1)=dU2(1,i+1)+U2(1,i);
dU(:,i+1)=[dU1(:,i+1);dU2(:,i+1)];

Y_m(:,i+1)=G*dU(:,i+1)+Y_past(:,i+1);

dU1_(2:n,i+2) = dU1_(1:n-1,i+1);
dU1_(1,i+2)=dU1(1,i+1);
dU2_(2:n,i+2) = dU2_(1:n-1,i+1);
dU2_(1,i+2)=dU2(1,i+1);
dU_(:,i+2)=[dU1_(:,i+2);dU2_(:,i+2)];
%Y_(2:n,i+2+miu(1))=Y_(1:n-1,i+1+miu(1)); % without using y(t+1) we can make Y_past in this way
% Y_(1,i+2+miu(1))=Y_m(1,i+1);

u1=U1(1,i+1);
u2=U2(1,i+1);
sim('Model');
d1(i+2)=y(end)-yl(end);
Y_(2:n,i+2)=Y_(1:n-1,i+1);
Y_(1,i+2)=yl(end);
y=y(end);%+dist(i,1); % nonlinear
y1=[y1; y+441.2];
ym=[ym; yl(end)];
u_1=[u_1; u1];
u_2=[u_2; u2];
%noise=[noise; n];
x01=x1(end);
x02=x2(end);

end
figure(3);
subplot(2,2,1:2);
plot(y1,'b');
grid on
title('Response of the nonlinear system');
xlabel('sample');
subplot(2,2,3);
plot(u_1,'b');
grid on
xlabel('sample');
title('Control law for input 1 without bias');
subplot(2,2,4);
plot(u_2,'b');
grid on
xlabel('sample');
title('Control law for input 2 without bias');
% .....
% ..... Q=60I.....
gamma =1;
gain_DC=(num1(1)+num1(2)+num1(3))/(den1(1)+den1(2)+den1(3));
gain_DC2=(num2(1)+num2(2)+num2(3))/(den2(1)+den2(2)+den2(3));
Q = 60*eye(2);
R1 =((1.2)^2)*gamma*gain_DC^2*eye(2);
R2=gamma*gain_DC2^2*eye(2);

```



```

R=[R1 zeros(2); zeros(2) R2];
alpha=0.5;
%for q=1:length(t)-100
    Kpfc=(G'*Q*G+R)\(G'*Q);
%end
%.....
n=3;
x01=0.0882;
x02=441.2;
dU1_=zeros(n,length(t)-100);
dU2_=zeros(n,length(t)-100);
dU_=[dU1_;dU2_];
d1=zeros(1,length(t)-100);
y1=441.2;
u_1=[];
u_2=[];
ym=[];
y=0;
Y_d=zeros(2,length(t)-100);
Y_past=zeros(2,length(t)-100);
Y_m=zeros(2,length(t)-100);
D=zeros(2,length(t)-100);
E=zeros(2,length(t)-100);
dU1=zeros(2,length(t)-100);
dU2=zeros(2,length(t)-100);
dU=[dU1;dU2];
U1=zeros(M,length(t)-100);
U2=zeros(M,length(t)-100);
Y_=zeros(n,length(t)-100);
%.....step.....
r =ones(length(t)-100,1);
%.....
for i=1:length(t)-101

for j=1:2
    Y_d(j,i+1)=(alpha^j)*y+(1-(alpha)^j)*r(i+1); % Programmed
end

Y_past(:,i+1)=M_(:,i+1)*dU_(:,i+1)+F(:,i+1)*Y_(:,i+1);
D(:,i+1)=d1(i+1)*ones(2,1);

E(:,i+1)=Y_d(:,i+1)-Y_past(:,i+1)-D(:,i+1);

dU(:,i+1)=Kpfc*E(:,i+1);
dU1(:,i+1)=dU(1:2,i+1);
dU2(:,i+1)=dU(3:4,i+1);
U1(1,i+1)=dU1(1,i+1)+U1(1,i);
U2(1,i+1)=dU2(1,i+1)+U2(1,i);
dU(:,i+1)=[dU1(:,i+1);dU2(:,i+1)];

Y_m(:,i+1)=G*dU(:,i+1)+Y_past(:,i+1);

dU1_(2:n,i+2) = dU1_(1:n-1,i+1);
dU1_(1,i+2)=dU1(1,i+1);
dU2_(2:n,i+2) = dU2_(1:n-1,i+1);

```

```

dU2_(1,i+2)=dU2(1,i+1);
dU_(:,i+2)=[dU1_(:,i+2);dU2_(:,i+2)];
%Y_(2:n,i+2+miu(1))=Y_(1:n-1,i+1+miu(1)); % without using y(t+1) we can make Y_past in this way
% Y_(1,i+2+miu(1))=Y_m(1,i+1);

u1=U1(1,i+1);
u2=U2(1,i+1);
sim('Model');
d1(i+2)=y(end)-yl(end);
Y_(2:n,i+2)=Y_(1:n-1,i+1);
Y_(1,i+2)=yl(end);
y=y(end);%+dist(i,1); % nonlinear
y1=[y1; y+441.2];
ym=[ym; yl(end)];
u_1=[u_1; u1];
u_2=[u_2; u2];
%noise=[noise; n];
x01=x1(end);
x02=x2(end);

end
figure(3);
subplot(2,2,1:2);
hold on
plot(y1,'c');
subplot(2,2,3);
hold on
plot(u_1,'c');
subplot(2,2,4);
hold on
plot(u_2,'c');
%.....
%..... Q=160I.....
gamma =1;
gain_DC=(num1(1)+num1(2)+num1(3))/(den1(1)+den1(2)+den1(3));
gain_DC2=(num2(1)+num2(2)+num2(3))/(den2(1)+den2(2)+den2(3));
Q =160*eye(2);
R1 =((1.2)^2)*gamma*gain_DC^2*eye(2);
R2=gamma*gain_DC2^2*eye(2);
R=[R1 zeros(2); zeros(2) R2];
alpha=0.5;
%for q=1:length(t)-100
    Kpfc=(G'*Q*G+R)\(G'*Q);
%end
%.....
n=3;
x01=0.0882;
x02=441.2;
dU1_=zeros(n,length(t)-100);
dU2_=zeros(n,length(t)-100);
dU_=[dU1_;dU2_];
d1=zeros(1,length(t)-100);
y1=441.2;
u_1=[];
u_2=[];

```

```

ym=[];
y=0;
Y_d=zeros(2,length(t)-100);
Y_past=zeros(2,length(t)-100);
Y_m=zeros(2,length(t)-100);
D=zeros(2,length(t)-100);
E=zeros(2,length(t)-100);
dU1=zeros(2,length(t)-100);
dU2=zeros(2,length(t)-100);
dU=[dU1;dU2];
U1=zeros(M,length(t)-100);
U2=zeros(M,length(t)-100);
Y_=zeros(n,length(t)-100);
%.....step.....
r =ones(length(t)-100,1);
%.....
for i=1:length(t)-101

for j=1:2
    Y_d(j,i+1)=(alpha^j)*y+(1-(alpha)^j)*r(i+1); % Programmed
end

Y_past(:,i+1)=M(:,i+1)*dU(:,i+1)+F(:,i+1)*Y_(:,i+1);
D(:,i+1)=d1(i+1)*ones(2,1);

E(:,i+1)=Y_d(:,i+1)-Y_past(:,i+1)-D(:,i+1);

dU(:,i+1)=Kpfc*E(:,i+1);
dU1(:,i+1)=dU(1:2,i+1);
dU2(:,i+1)=dU(3:4,i+1);
U1(1,i+1)=dU1(1,i+1)+U1(1,i);
U2(1,i+1)=dU2(1,i+1)+U2(1,i);
dU(:,i+1)=[dU1(:,i+1);dU2(:,i+1)];

Y_m(:,i+1)=G*dU(:,i+1)+Y_past(:,i+1);

dU1_(2:n,i+2) = dU1_(1:n-1,i+1);
dU1_(1,i+2)=dU1(1,i+1);
dU2_(2:n,i+2) = dU2_(1:n-1,i+1);
dU2_(1,i+2)=dU2(1,i+1);
dU_(:,i+2)=[dU1_(:,i+2);dU2_(:,i+2)];
% Y_(2:n,i+2+miu(1))=Y_(1:n-1,i+1+miu(1)); % without using y(t+1) we can make Y_past in this way
% Y_(1,i+2+miu(1))=Y_m(1,i+1);

u1=U1(1,i+1);
u2=U2(1,i+1);
sim('Model');
d1(i+2)=y(end)-yl(end);
Y_(2:n,i+2)=Y_(1:n-1,i+1);
Y_(1,i+2)=yl(end);
y=y(end);%+dist(i,1); % nonlinear
yl=[y1; y+441.2];
ym=[ym; yl(end)];
u_1=[u_1; u1];

```

```

u_2=[u_2; u2];
%noise=[noise; n];
x01=x1(end);
x02=x2(end);

end
figure(3);
subplot(2,2,1:2);
hold on
plot(y1,'m');
hold on
plot(r+441.2,'r');
legend('Q=I','Q=60I','Q=160I','r');

```

```

subplot(2,2,3);
hold on
plot(u_1,'m');
legend('Q=I','Q=60I','Q=160I');
subplot(2,2,4);
hold on
plot(u_2,'m');
legend('Q=I','Q=60I','Q=160I');

```

```

% veriying alpha.....
%.....
clear
clc
[n1,d1,n2,d2]=Inputsys(1);
Gs1 = tf(n1,d1);
Ts=0.1;
Gd1 = c2d(Gs1,Ts,'zoh');
[num1,den1]=tfdata(Gd1,'v');
Gs2 = tf(n2,d2);
Gd2 = c2d(Gs2,Ts,'zoh');
[num2,den2]=tfdata(Gd2,'v');
sys_info = stepinfo(Gd1);
ts1 = sys_info.SettlingTime;
tr1=sys_info.RiseTime;
sys_info = stepinfo(Gd2);
ts2 = sys_info.SettlingTime;
tr2=sys_info.RiseTime;
t=1:Ts:80;
[g1,t1] = step(Gd1,t);
[g2,t2] = step(Gd2,t);
P1=floor(tr1/Ts);
P2=floor(tr2/Ts);
N1=floor( ts1/Ts);
N2=floor( ts2/Ts);
P=max(P1,P2);
N=max(N1,N2);
M=P;
n=3;
m1=2;
miu=[3 5];
ui1=randn(1,length(t));

```

```

ui2=randn(1,length(t));
ui=[ui1; ui2];
Gd=[Gd1 Gd2];
[yi,t1]=lsim(Gd,ui,t);
theta1=zeros(2*miu(1)+3*n,length(t)-100);
theta2=zeros(2*miu(2)+3*n,length(t)-100);
for i=1:2
    phi1=zeros(2*miu(i)+3*n,length(t));
    for j=1:length(t)-100
        y(1,j)=yi(j+miu(i));
        for k=1:miu(i)
            if (j+miu(i)-k-1)<=0
                du(j+miu(i)-k)=ui1(j+miu(i)-k);
                du2(j+miu(i)-k)=ui2(j+miu(i)-k);
            else
                du(j+miu(i)-k)=ui1(j+miu(i)-k)-ui1(j+miu(i)-k-1);
                du2(j+miu(i)-k)=ui2(j+miu(i)-k)-ui2(j+miu(i)-k-1);
            end
            phi1(k,j)=du(j+miu(i)-k);
            phi1(k+miu(i),j)=du2(j+miu(i)-k);
        end
        for k=1:n
            if (j-k-1)<=0
                phi1(k+2*miu(i),j)=0;
                phi1(k+2*miu(i)+n,j)=0;
            else
                phi1(k+2*miu(i),j)=ui1(j-k)-ui1(j-k-1);
                phi1(k+2*miu(i)+n,j)=ui2(j-k)-ui2(j-k-1);
            end
            if (j-k)<=0
                phi1(k+2*miu(i)+2*n,j)=0;
            else
                phi1(k+2*miu(i)+2*n,j)=yi(j-k);
            end
        end
        if i==1
            %theta1(:,j)=(phi1(:,j)*phi1(:,j))\'\'(phi1(:,j)*y(1,j)); % we could use this method (right inverse)
            theta1(:,j)=pinv(phi1(:,j))\'\'*y(1,j);
        else
            %theta2(:,j)=(phi1(:,j)*phi1(:,j))\'\'(phi1(:,j)*y(1,j)); % we could use this method (right inverse)
            theta2(:,j)=pinv(phi1(:,j))\'\'*y(1,j);
        end
    end
end
%.....
G1i=zeros(2,2,length(t)-100); % these comments are related to the state which we use IIR for the future
G2i=zeros(2,2,length(t)-100);
M1_=zeros(2,n,length(t)-100);
M2_=zeros(2,n,length(t)-100);
F=zeros(2,n,length(t)-100);
for q=1:length(t)-100
    G1i(:,q)=[theta1(miu(1),q) theta1(miu(1)-m1,q); theta2(miu(2),q) theta2(miu(2)-m1,q)];
    G2i(:,q)=[theta1(2*miu(1),q) theta1(2*miu(1)-m1,q); theta2(2*miu(2),q) theta2(2*miu(2)-m1,q)];
    M1_(:,q)=[theta1(2*miu(1)+1:n+2*miu(1),q)'; theta2(2*miu(2)+1:n+2*miu(2),q)'];
    M2_(:,q)=[theta1(2*miu(1)+1+n:2*n+2*miu(1),q)'; theta2(2*miu(2)+n+1:2*n+2*miu(2),q)'];
    F(:,q)=[theta1(2*miu(1)+2*n+1:3*n+2*miu(1),q)'; theta2(2*miu(2)+2*n+1:2*miu(2)+3*n,q)'];
end

```

```

end
Gi=[G1i G2i];
%.....Toeplitz Matrix.....
b1 = zeros(1,P); b1(1,1)= g1(2);
a1 = g1(2:P+1);
G1 = toeplitz(a1,b1);
G1(:,M) = G1(:,M:P)*ones(P-M+1,1);
G1 = G1(:,1:M);
%.....
b2 = zeros(1,P); b2(1,1)= g2(2);
a2 = g2(2:P+1);
G2 = toeplitz(a2,b2);
G2(:,M) = G2(:,M:P)*ones(P-M+1,1);
G2 = G2(:,1:M);
%.....
G11=[G1(miu(1),1) G1(miu(1),m1+1); G1(miu(2),1) G1(miu(2),m1+1)];
G21=[G2(miu(1),1) G2(miu(1),m1+1); G2(miu(2),1) G2(miu(2),m1+1)];
G=[G11 G21];
M_=[M1_ M2_];
%.....
%..... alpha=0.4.....
gamma =1/60;
gain_DC=(num1(1)+num1(2)+num1(3))/(den1(1)+den1(2)+den1(3));
gain_DC2=(num2(1)+num2(2)+num2(3))/(den2(1)+den2(2)+den2(3));
Q = eye(2);
R1 =((1.2)^2)*gamma*gain_DC^2*eye(2);
R2=gamma*gain_DC2^2*eye(2);
R=[R1 zeros(2); zeros(2) R2];
alpha=0.4;
%for q=1:length(t)-100
    Kpfc=(G'*Q*G+R)\(G'*Q);
%end
%.....

x01=0.0882;
x02=441.2;
dU1_=zeros(n,length(t)-100);
dU2_=zeros(n,length(t)-100);
dU_=[dU1_;dU2_];
d1=zeros(1,length(t)-100);
y1=441.2;
u_1=[];
u_2=[];
ym=[];
y=0;
Y_d=zeros(2,length(t)-100);
Y_past=zeros(2,length(t)-100);
Y_m=zeros(2,length(t)-100);
D=zeros(2,length(t)-100);
E=zeros(2,length(t)-100);
dU1=zeros(2,length(t)-100);
dU2=zeros(2,length(t)-100);
dU=[dU1;dU2];
U1=zeros(M,length(t)-100);
U2=zeros(M,length(t)-100);
Y_=zeros(n,length(t)-100);

```

```

% .....step.....
r =ones(length(t)-100,1);
% .....
for i=1:length(t)-101

for j=1:2
    Y_d(j,i+1)=(alpha^j)*y+(1-(alpha)^j)*r(i+1); % Programmed
end

Y_past(:,i+1)=M_(:,i+1)*dU_(:,i+1)+F(:,i+1)*Y_(:,i+1);
D(:,i+1)=d1(i+1)*ones(2,1);

E(:,i+1)=Y_d(:,i+1)-Y_past(:,i+1)-D(:,i+1);

dU(:,i+1)=Kpfc*E(:,i+1);
dU1(:,i+1)=dU(1:2,i+1);
dU2(:,i+1)=dU(3:4,i+1);
U1(1,i+1)=dU1(1,i+1)+U1(1,i);
U2(1,i+1)=dU2(1,i+1)+U2(1,i);
dU(:,i+1)=[dU1(:,i+1);dU2(:,i+1)];

Y_m(:,i+1)=G*dU(:,i+1)+Y_past(:,i+1);

dU1_(2:n,i+2) = dU1_(1:n-1,i+1);
dU1_(1,i+2)=dU1(1,i+1);
dU2_(2:n,i+2) = dU2_(1:n-1,i+1);
dU2_(1,i+2)=dU2(1,i+1);
dU_(:,i+2)=[dU1_(:,i+2);dU2_(:,i+2)];
% Y_(2:n,i+2+miu(1))=Y_(1:n-1,i+1+miu(1)); % without using y(t+1) we can make Y_past in this way
% Y_(1,i+2+miu(1))=Y_m(1,i+1);

u1=U1(1,i+1);
u2=U2(1,i+1);
sim('Model');
d1(i+2)=y(end)-yl(end);
Y_(2:n,i+2)=Y_(1:n-1,i+1);
Y_(1,i+2)=yl(end);
y=y(end);%+dist(i,1); % nonlinear
y1=[y1; y+441.2];
ym=[ym; yl(end)];
u_1=[u_1; u1];
u_2=[u_2; u2];
%noise=[noise; n];
x01=x1(end);
x02=x2(end);

end
figure(3);
subplot(2,2,1:2);
plot(y1,'b');
grid on
title('Response of the nonlinear system');
xlabel('sample');
subplot(2,2,3);

```

```

plot(u_1,'b');
grid on
xlabel('sample');
title('Control law for input 1 without bias');
subplot(2,2,4);
plot(u_2,'b');
grid on
xlabel('sample');
title('Control law for input 2 without bias');
%.....
%..... alpha=0.6.....
gamma =1/60;
gain_DC=(num1(1)+num1(2)+num1(3))/(den1(1)+den1(2)+den1(3));
gain_DC2=(num2(1)+num2(2)+num2(3))/(den2(1)+den2(2)+den2(3));
Q = eye(2);
R1 =((1.2)^2)*gamma*gain_DC^2*eye(2);
R2=gamma*gain_DC2^2*eye(2);
R=[R1 zeros(2); zeros(2) R2];
alpha=0.6;
%for q=1:length(t)-100
    Kpfc=(G'*Q*G+R)\(G'*Q);
%end
%.....
n=3;
x01=0.0882;
x02=441.2;
dU1_=zeros(n,length(t)-100);
dU2_=zeros(n,length(t)-100);
dU_=[dU1_;dU2_];
d1=zeros(1,length(t)-100);
y1=441.2;
u_1=[];
u_2=[];
ym=[];
y=0;
Y_d=zeros(2,length(t)-100);
Y_past=zeros(2,length(t)-100);
Y_m=zeros(2,length(t)-100);
D=zeros(2,length(t)-100);
E=zeros(2,length(t)-100);
dU1=zeros(2,length(t)-100);
dU2=zeros(2,length(t)-100);
dU=[dU1;dU2];
U1=zeros(M,length(t)-100);
U2=zeros(M,length(t)-100);
Y_=zeros(n,length(t)-100);
%.....step.....
r =ones(length(t)-100,1);
%.....
for i=1:length(t)-101

for j=1:2
    Y_d(j,i+1)=(alpha^j)*y+(1-(alpha)^j)*r(i+1); % Programmed
end

```



```

Y_past(:,i+1)=M_(:,i+1)*dU_(:,i+1)+F(:,i+1)*Y_(:,i+1);
D(:,i+1)=d1(i+1)*ones(2,1);

E(:,i+1)=Y_d(:,i+1)-Y_past(:,i+1)-D(:,i+1);

dU(:,i+1)=Kpfc*E(:,i+1);
dU1(:,i+1)=dU(1:2,i+1);
dU2(:,i+1)=dU(3:4,i+1);
U1(1,i+1)=dU1(1,i+1)+U1(1,i);
U2(1,i+1)=dU2(1,i+1)+U2(1,i);
dU(:,i+1)=[dU1(:,i+1);dU2(:,i+1)];

Y_m(:,i+1)=G*dU(:,i+1)+Y_past(:,i+1);

dU1_(2:n,i+2) = dU1_(1:n-1,i+1);
dU1_(1,i+2)=dU1(1,i+1);
dU2_(2:n,i+2) = dU2_(1:n-1,i+1);
dU2_(1,i+2)=dU2(1,i+1);
dU_(:,i+2)=[dU1_(:,i+2);dU2_(:,i+2)];
% Y_(2:n,i+2+miu(1))=Y_(1:n-1,i+1+miu(1)); % without using y(t+1) we can make Y_past in this way
% Y_(1,i+2+miu(1))=Y_m(1,i+1);

u1=U1(1,i+1);
u2=U2(1,i+1);
sim('Model');
d1(i+2)=y(end)-yl(end);
Y_(2:n,i+2)=Y_(1:n-1,i+1);
Y_(1,i+2)=yl(end);
y=y(end);%+dist(i,1); % nonlinear
y1=[y1; y+441.2];
ym=[ym; yl(end)];
u_1=[u_1; u1];
u_2=[u_2; u2];
%noise=[noise; n];
x01=x1(end);
x02=x2(end);

end
figure(3);
subplot(2,2,1:2);
hold on
plot(y1,'c');
subplot(2,2,3);
hold on
plot(u_1,'c');
subplot(2,2,4);
hold on
plot(u_2,'c');
% .....
% ..... alpha=0.8 .....
gamma =1/60;
gain_DC=(num1(1)+num1(2)+num1(3))/(den1(1)+den1(2)+den1(3));
gain_DC2=(num2(1)+num2(2)+num2(3))/(den2(1)+den2(2)+den2(3));
Q =eye(2);
R1 =((1.2)^2)*gamma*gain_DC^2*eye(2);

```

```

R2=gamma*gain_DC2^2*eye(2);
R=[R1 zeros(2); zeros(2) R2];
alpha=0.8;
%for q=1:length(t)-100
    Kpfc=(G'*Q*G+R)/(G'*Q);
%end
%.....
n=3;
x01=0.0882;
x02=441.2;
dU1_=zeros(n,length(t)-100);
dU2_=zeros(n,length(t)-100);
dU_=[dU1_;dU2_];
d1=zeros(1,length(t)-100);
y1=441.2;
u_1=[];
u_2=[];
ym=[];
y=0;
Y_d=zeros(2,length(t)-100);
Y_past=zeros(2,length(t)-100);
Y_m=zeros(2,length(t)-100);
D=zeros(2,length(t)-100);
E=zeros(2,length(t)-100);
dU1=zeros(2,length(t)-100);
dU2=zeros(2,length(t)-100);
dU=[dU1;dU2];
U1=zeros(M,length(t)-100);
U2=zeros(M,length(t)-100);
Y_=zeros(n,length(t)-100);
%.....step.....
r =ones(length(t)-100,1);
%.....
for i=1:length(t)-101

for j=1:2
    Y_d(j,i+1)=(alpha^j)*y+(1-(alpha)^j)*r(i+1); % Programmed
end

Y_past(:,i+1)=M_(:,i+1)*dU_(:,i+1)+F(:,i+1)*Y_(:,i+1);
D(:,i+1)=d1(i+1)*ones(2,1);

E(:,i+1)=Y_d(:,i+1)-Y_past(:,i+1)-D(:,i+1);

dU(:,i+1)=Kpfc*E(:,i+1);
dU1(:,i+1)=dU(1:2,i+1);
dU2(:,i+1)=dU(3:4,i+1);
U1(1,i+1)=dU1(1,i+1)+U1(1,i);
U2(1,i+1)=dU2(1,i+1)+U2(1,i);
dU(:,i+1)=[dU1(:,i+1);dU2(:,i+1)];

Y_m(:,i+1)=G*dU(:,i+1)+Y_past(:,i+1);

dU1_(2:n,i+2) = dU1_(1:n-1,i+1);
dU1_(1,i+2)=dU1(1,i+1);

```

```

dU2_(2:n,i+2) = dU2_(1:n-1,i+1);
dU2_(1,i+2)=dU2(1,i+1);
dU_(:,i+2)=[dU1_(:,i+2);dU2_(:,i+2)];
%Y_(2:n,i+2+miu(1))=Y_(1:n-1,i+1+miu(1)); % without using y(t+1) we can make Y_past in this way
% Y_(1,i+2+miu(1))=Y_m(1,i+1);

u1=U1(1,i+1);
u2=U2(1,i+1);
sim('Model');
d1(i+2)=y(end)-yl(end);
Y_(2:n,i+2)=Y_(1:n-1,i+1);
Y_(1,i+2)=yl(end);
y=y(end);%+dist(i,1); % nonlinear
y1=[y1; y+441.2];
ym=[ym; yl(end)];
u_1=[u_1; u1];
u_2=[u_2; u2];
%noise=[noise; n];
x01=x1(end);
x02=x2(end);

end
figure(3);
subplot(2,2,1:2);
hold on
plot(y1,'m');
hold on
plot(r+441.2,'r');
legend('alpha=0.4','alpha=0.6','alpha=0.8','r');
subplot(2,2,3);
hold on
plot(u_1,'m');
legend('alpha=0.4','alpha=0.6','alpha=0.8');
subplot(2,2,4);
hold on
plot(u_2,'m');
legend('alpha=0.4','alpha=0.6','alpha=0.8');

% veriyng Ts.....
% .....
% Ts=0.05
clear
clc
[n1,d1,n2,d2]=Inputsys(1);
Gs1 = tf(n1,d1);
Ts=0.05;
Gd1 = c2d(Gs1,Ts,'zoh');
[num1,den1]=tfdata(Gd1,'v');
Gs2 = tf(n2,d2);
Gd2 = c2d(Gs2,Ts,'zoh');
[num2,den2]=tfdata(Gd2,'v');
sys_info = stepinfo(Gd1);
ts1 = sys_info.SettlingTime;
tr1=sys_info.RiseTime;
sys_info = stepinfo(Gd2);

```

```

ts2 = sys_info.SettlingTime;
tr2=sys_info.RiseTime;
t=1:Ts:40;
[g1,t1] = step(Gd1,t);
[g2,t2] = step(Gd2,t);
P1=floor(tr1/Ts);
P2=floor(tr2/Ts);
N1=floor( ts1/Ts);
N2=floor( ts2/Ts);
P=max(P1,P2);
N=max(N1,N2);
M=P;
n=3;
m1=2;
miu=[3 5];
ui1=randn(1,length(t));
ui2=randn(1,length(t));
ui=[ui1; ui2];
Gd=[Gd1 Gd2];
[yi,t1]=lsim(Gd,ui,t);
theta1=zeros(2*miu(1)+3*n,length(t)-100);
theta2=zeros(2*miu(2)+3*n,length(t)-100);
for i=1:2
    phi1=zeros(2*miu(i)+3*n,length(t));
    for j=1:length(t)-100
        y(1,j)=yi(j+miu(i));
        for k=1:miu(i)
            if (j+miu(i)-k-1)<=0
                du(j+miu(i)-k)=ui1(j+miu(i)-k);
                du2(j+miu(i)-k)=ui2(j+miu(i)-k);
            else
                du(j+miu(i)-k)=ui1(j+miu(i)-k)-ui1(j+miu(i)-k-1);
                du2(j+miu(i)-k)=ui2(j+miu(i)-k)-ui2(j+miu(i)-k-1);
            end
            phi1(k,j)=du(j+miu(i)-k);
            phi1(k+miu(i),j)=du2(j+miu(i)-k);
        end
        for k=1:n
            if (j-k-1)<=0
                phi1(k+2*miu(i),j)=0;
                phi1(k+2*miu(i)+n,j)=0;
            else
                phi1(k+2*miu(i),j)=ui1(j-k)-ui1(j-k-1);
                phi1(k+2*miu(i)+n,j)=ui2(j-k)-ui2(j-k-1);
            end
            if (j-k)<=0
                phi1(k+2*miu(i)+2*n,j)=0;
            else
                phi1(k+2*miu(i)+2*n,j)=yi(j-k);
            end
        end
    end
    if i==1
        %theta1(:,j)=(phi1(:,j)*phi1(:,j))\'\'(phi1(:,j)*y(1,j)); % we could use this method (right inverse)
        theta1(:,j)=pinv(phi1(:,j))\'\'*y(1,j);
    else
        %theta2 (:,j)=(phi1(:,j)*phi1(:,j))\'\'(phi1(:,j)*y(1,j)); % we could use this method (right inverse)
    end
end

```

```

        theta2(:,j)=pinv(phi1(:,j))*y(1,j);
    end
end
end
%.....
G1i=zeros(2,2,length(t)-100); % these comments are related to the state which we use IIR for the future
G2i=zeros(2,2,length(t)-100);
M1_=zeros(2,n,length(t)-100);
M2_=zeros(2,n,length(t)-100);
F=zeros(2,n,length(t)-100);
for q=1:length(t)-100
    G1i(:,q)=[theta1(miu(1),q) theta1(miu(1)-m1,q); theta2(miu(2),q) theta2(miu(2)-m1,q)];
    G2i(:,q)=[theta1(2*miu(1),q) theta1(2*miu(1)-m1,q); theta2(2*miu(2),q) theta2(2*miu(2)-m1,q)];
    M1_(:,q)=[theta1(2*miu(1)+1:n+2*miu(1),q)'; theta2(2*miu(2)+1:n+2*miu(2),q)'];
    M2_(:,q)=[theta1(2*miu(1)+1+n:2*n+2*miu(1),q)'; theta2(2*miu(2)+n+1:2*n+2*miu(2),q)'];
    F(:,q)=[theta1(2*miu(1)+2*n+1:3*n+2*miu(1),q)'; theta2(2*miu(2)+2*n+1:2*miu(2)+3*n,q)'];
end
Gi=[G1i G2i];
%.....Toeplitz Matrix.....
b1 = zeros(1,P); b1(1,1)= g1(2);
a1 = g1(2:P+1);
G1 = toeplitz(a1,b1);
G1(:,M) = G1(:,M:P)*ones(P-M+1,1);
G1 = G1(:,1:M);
%.....
b2 = zeros(1,P); b2(1,1)= g2(2);
a2 = g2(2:P+1);
G2 = toeplitz(a2,b2);
G2(:,M) = G2(:,M:P)*ones(P-M+1,1);
G2 = G2(:,1:M);
%.....
G11=[G1(miu(1),1) G1(miu(1),m1+1); G1(miu(2),1) G1(miu(2),m1+1)];
G21=[G2(miu(1),1) G2(miu(1),m1+1); G2(miu(2),1) G2(miu(2),m1+1)];
G=[G11 G21];
M_=[M1_ M2_];
%.....
gamma =1/60;
gain_DC=(num1(1)+num1(2)+num1(3))/(den1(1)+den1(2)+den1(3));
gain_DC2=(num2(1)+num2(2)+num2(3))/(den2(1)+den2(2)+den2(3));
Q = eye(2);
R1 =((1.2)^2)*gamma*gain_DC^2*eye(2);
R2=gamma*gain_DC2^2*eye(2);
R=[R1 zeros(2); zeros(2) R2];
alpha=0.5;
%for q=1:length(t)-100
    Kpfc=(G'*Q*G+R)\(G'*Q);
%end
%.....

x01=0.0882;
x02=441.2;
dU1_=zeros(n,length(t)-100);
dU2_=zeros(n,length(t)-100);
dU_=[dU1_;dU2_];
d1=zeros(1,length(t)-100);
y1=441.2;

```

```

u_1=[];
u_2=[];
ym=[];
y=0;
Y_d=zeros(2,length(t)-100);
Y_past=zeros(2,length(t)-100);
Y_m=zeros(2,length(t)-100);
D=zeros(2,length(t)-100);
E=zeros(2,length(t)-100);
dU1=zeros(2,length(t)-100);
dU2=zeros(2,length(t)-100);
dU=[dU1;dU2];
U1=zeros(M,length(t)-100);
U2=zeros(M,length(t)-100);
Y_=zeros(n,length(t)-100);
%.....step.....
r =ones(length(t)-100,1);
%.....
for i=1:length(t)-101

for j=1:2
    Y_d(j,i+1)=(alpha^j)*y+(1-(alpha)^j)*r(i+1); % Programmed
end

Y_past(:,i+1)=M(:,i+1)*dU(:,i+1)+F(:,i+1)*Y_(:,i+1);
D(:,i+1)=d1(i+1)*ones(2,1);

E(:,i+1)=Y_d(:,i+1)-Y_past(:,i+1)-D(:,i+1);

dU(:,i+1)=Kpfc*E(:,i+1);
dU1(:,i+1)=dU(1:2,i+1);
dU2(:,i+1)=dU(3:4,i+1);
U1(1,i+1)=dU1(1,i+1)+U1(1,i);
U2(1,i+1)=dU2(1,i+1)+U2(1,i);
dU(:,i+1)=[dU1(:,i+1);dU2(:,i+1)];

Y_m(:,i+1)=G*dU(:,i+1)+Y_past(:,i+1);

dU1_(2:n,i+2) = dU1_(1:n-1,i+1);
dU1_(1,i+2)=dU1(1,i+1);
dU2_(2:n,i+2) = dU2_(1:n-1,i+1);
dU2_(1,i+2)=dU2(1,i+1);
dU_(:,i+2)=[dU1_(:,i+2);dU2_(:,i+2)];
%Y_(2:n,i+2+miu(1))=Y_(1:n-1,i+1+miu(1)); % without using y(t+1) we can make Y_past in this way
% Y_(1,i+2+miu(1))=Y_m(1,i+1);

u1=U1(1,i+1);
u2=U2(1,i+1);
sim('Model');
d1(i+2)=y(end)-y1(end);
Y_(2:n,i+2)=Y_(1:n-1,i+1);
Y_(1,i+2)=y1(end);
y=y(end);%+dist(i,1); % nonlinear
y1=[y1; y+441.2];

```

```

ym=[ym; yl(end)];
u_1=[u_1; u1];
u_2=[u_2; u2];
%noise=[noise; n];
x01=x1(end);
x02=x2(end);
end
figure(3);
subplot(2,2,1:2);
plot(y1,'b');
grid on
title('Response of the nonlinear system');
xlabel('sample');
subplot(2,2,3);
plot(u_1,'b');
grid on
xlabel('sample');
title('Control law for input 1 without bias');
subplot(2,2,4);
plot(u_2,'b');
grid on
xlabel('sample');
title('Control law for input 2 without bias');
% .....
% Ts=0.1
clear
clc
[n1,d1,n2,d2]=Inputsys(1);
Gs1 = tf(n1,d1);
Ts=0.1;
Gd1 = c2d(Gs1,Ts,'zoh');
[num1,den1]=tfdata(Gd1,'v');
Gs2 = tf(n2,d2);
Gd2 = c2d(Gs2,Ts,'zoh');
[num2,den2]=tfdata(Gd2,'v');
sys_info = stepinfo(Gd1);
ts1 = sys_info.SettlingTime;
tr1=sys_info.RiseTime;
sys_info = stepinfo(Gd2);
ts2 = sys_info.SettlingTime;
tr2=sys_info.RiseTime;
t=1:Ts:80;
[g1,t1] = step(Gd1,t);
[g2,t2] = step(Gd2,t);
P1=floor(tr1/Ts);
P2=floor(tr2/Ts);
N1=floor( ts1/Ts);
N2=floor( ts2/Ts);
P=max(P1,P2);
N=max(N1,N2);
M=P;
n=3;
m1=2;
miu=[3 5];
ui1=randn(1,length(t));
ui2=randn(1,length(t));

```

```

ui=[ui1; ui2];
Gd=[Gd1 Gd2];
[yi,t1]=lsim(Gd,ui,t);
theta1=zeros(2*miu(1)+3*n,length(t)-100);
theta2=zeros(2*miu(2)+3*n,length(t)-100);
for i=1:2
    phi1=zeros(2*miu(i)+3*n,length(t));
    for j=1:length(t)-100
        y(1,j)=yi(j+miu(i));
        for k=1:miu(i)
            if (j+miu(i)-k-1)<=0
                du(j+miu(i)-k)=ui1(j+miu(i)-k);
                du2(j+miu(i)-k)=ui2(j+miu(i)-k);
            else
                du(j+miu(i)-k)=ui1(j+miu(i)-k)-ui1(j+miu(i)-k-1);
                du2(j+miu(i)-k)=ui2(j+miu(i)-k)-ui2(j+miu(i)-k-1);
            end
            phi1(k,j)=du(j+miu(i)-k);
            phi1(k+miu(i),j)=du2(j+miu(i)-k);
        end
        for k=1:n
            if (j-k-1)<=0
                phi1(k+2*miu(i),j)=0;
                phi1(k+2*miu(i)+n,j)=0;
            else
                phi1(k+2*miu(i),j)=ui1(j-k)-ui1(j-k-1);
                phi1(k+2*miu(i)+n,j)=ui2(j-k)-ui2(j-k-1);
            end
            if (j-k)<=0
                phi1(k+2*miu(i)+2*n,j)=0;
            else
                phi1(k+2*miu(i)+2*n,j)=yi(j-k);
            end
        end
    end
    if i==1
        %theta1(:,j)=(phi1(:,j)*phi1(:,j))\'\'(phi1(:,j)*y(1,j)); % we could use this method (right inverse)
        theta1(:,j)=pinv(phi1(:,j))\'\'*y(1,j);
    else
        %theta2(:,j)=(phi1(:,j)*phi1(:,j))\'\'(phi1(:,j)*y(1,j)); % we could use this method (right inverse)
        theta2(:,j)=pinv(phi1(:,j))\'\'*y(1,j);
    end
end
end
%.....
G1i=zeros(2,2,length(t)-100); % these comments are related to the state which we use IIR for the future
G2i=zeros(2,2,length(t)-100);
M1_=zeros(2,n,length(t)-100);
M2_=zeros(2,n,length(t)-100);
F=zeros(2,n,length(t)-100);
for q=1:length(t)-100
    G1i(:,q)=[theta1(miu(1),q) theta1(miu(1)-m1,q); theta2(miu(2),q) theta2(miu(2)-m1,q)];
    G2i(:,q)=[theta1(2*miu(1),q) theta1(2*miu(1)-m1,q); theta2(2*miu(2),q) theta2(2*miu(2)-m1,q)];
    M1_(:,q)=[theta1(2*miu(1)+1:n+2*miu(1),q)'; theta2(2*miu(2)+1:n+2*miu(2),q)'];
    M2_(:,q)=[theta1(2*miu(1)+1+n:2*n+2*miu(1),q)'; theta2(2*miu(2)+n+1:2*n+2*miu(2),q)'];
    F(:,q)=[theta1(2*miu(1)+2*n+1:3*n+2*miu(1),q)'; theta2(2*miu(2)+2*n+1:2*miu(2)+3*n,q)'];
end

```



```

Gi=[G1i G2i];
%.....Toeplitz Matrix.....
b1 = zeros(1,P); b1(1,1)= g1(2);
a1 = g1(2:P+1);
G1 = toeplitz(a1,b1);
G1(:,M) = G1(:,M:P)*ones(P-M+1,1);
G1 = G1(:,1:M);
%.....
b2 = zeros(1,P); b2(1,1)= g2(2);
a2 = g2(2:P+1);
G2 = toeplitz(a2,b2);
G2(:,M) = G2(:,M:P)*ones(P-M+1,1);
G2 = G2(:,1:M);
%.....
G11=[G1(miu(1),1) G1(miu(1),m1+1); G1(miu(2),1) G1(miu(2),m1+1)];
G21=[G2(miu(1),1) G2(miu(1),m1+1); G2(miu(2),1) G2(miu(2),m1+1)];
G=[G11 G21];
M_=[M1_ M2_];
%.....
gamma =1/60;
gain_DC=(num1(1)+num1(2)+num1(3))/(den1(1)+den1(2)+den1(3));
gain_DC2=(num2(1)+num2(2)+num2(3))/(den2(1)+den2(2)+den2(3));
Q = eye(2);
R1 =((1.2)^2)*gamma*gain_DC^2*eye(2);
R2=gamma*gain_DC2^2*eye(2);
R=[R1 zeros(2); zeros(2) R2];
alpha=0.5;
%for q=1:length(t)-100
    Kpfc=(G'*Q*G+R)/(G'*Q);
%end
%.....

x01=0.0882;
x02=441.2;
dU1_=zeros(n,length(t)-100);
dU2_=zeros(n,length(t)-100);
dU_=[dU1_;dU2_];
d1=zeros(1,length(t)-100);
y1=441.2;
u_1=[];
u_2=[];
ym=[];
y=0;
Y_d=zeros(2,length(t)-100);
Y_past=zeros(2,length(t)-100);
Y_m=zeros(2,length(t)-100);
D=zeros(2,length(t)-100);
E=zeros(2,length(t)-100);
dU1=zeros(2,length(t)-100);
dU2=zeros(2,length(t)-100);
dU=[dU1;dU2];
U1=zeros(M,length(t)-100);
U2=zeros(M,length(t)-100);
Y_=zeros(n,length(t)-100);
%.....step.....
r =ones(length(t)-100,1);

```

```

% .....
for i=1:length(t)-101

for j=1:2
    Y_d(j,i+1)=(alpha^j)*y+(1-(alpha)^j)*r(i+1); % Programmed
end

Y_past(:,i+1)=M_(:,i+1)*dU(:,i+1)+F(:,i+1)*Y_(:,i+1);
D(:,i+1)=d1(i+1)*ones(2,1);

E(:,i+1)=Y_d(:,i+1)-Y_past(:,i+1)-D(:,i+1);

dU(:,i+1)=Kpfc*E(:,i+1);
dU1(:,i+1)=dU(1:2,i+1);
dU2(:,i+1)=dU(3:4,i+1);
U1(1,i+1)=dU1(1,i+1)+U1(1,i);
U2(1,i+1)=dU2(1,i+1)+U2(1,i);
dU(:,i+1)=[dU1(:,i+1);dU2(:,i+1)];

Y_m(:,i+1)=G*dU(:,i+1)+Y_past(:,i+1);

dU1_(2:n,i+2) = dU1_(1:n-1,i+1);
dU1_(1,i+2)=dU1(1,i+1);
dU2_(2:n,i+2) = dU2_(1:n-1,i+1);
dU2_(1,i+2)=dU2(1,i+1);
dU_(:,i+2)=[dU1_(:,i+2);dU2_(:,i+2)];
% Y_(2:n,i+2+miu(1))=Y_(1:n-1,i+1+miu(1)); % without using y(t+1) we can make Y_past in this way
% Y_(1,i+2+miu(1))=Y_m(1,i+1);

u1=U1(1,i+1);
u2=U2(1,i+1);
sim('Model');
d1(i+2)=y(end)-y1(end);
Y_(2:n,i+2)=Y_(1:n-1,i+1);
Y_(1,i+2)=y1(end);
y=y(end);%+dist(i,1); % nonlinear
y1=[y1; y+441.2];
ym=[ym; y1(end)];
u_1=[u_1; u1];
u_2=[u_2; u2];
%noise=[noise; n];
x01=x1(end);
x02=x2(end);
end
figure(3);
subplot(2,2,1:2);
hold on
plot(y1,'c');
subplot(2,2,3);
hold on
plot(u_1,'c');
subplot(2,2,4);
hold on
plot(u_2,'c');

```

```

% .....
% Ts=0.5
clear
clc
[n1,d1,n2,d2]=Inputsys(1);
Gs1 = tf(n1,d1);
Ts=0.5;
Gd1 = c2d(Gs1,Ts,'zoh');
[num1,den1]=tfdata(Gd1,'v');
Gs2 = tf(n2,d2);
Gd2 = c2d(Gs2,Ts,'zoh');
[num2,den2]=tfdata(Gd2,'v');
sys_info = stepinfo(Gd1);
ts1 = sys_info.SettlingTime;
tr1=sys_info.RiseTime;
sys_info = stepinfo(Gd2);
ts2 = sys_info.SettlingTime;
tr2=sys_info.RiseTime;
t=1:Ts:400;
[g1,t1] = step(Gd1,t);
[g2,t2] = step(Gd2,t);
P1=floor(tr1/Ts);
P2=floor(tr2/Ts);
N1=floor( ts1/Ts);
N2=floor( ts2/Ts);
P=max(P1,P2);
N=max(N1,N2);
M=P;
n=3;
m1=2;
miu=[3 5];
ui1=randn(1,length(t));
ui2=randn(1,length(t));
ui=[ui1; ui2];
Gd=[Gd1 Gd2];
[yi,t1]=lsim(Gd,ui,t);
theta1=zeros(2*miu(1)+3*n,length(t)-100);
theta2=zeros(2*miu(2)+3*n,length(t)-100);
for i=1:2
    phi1=zeros(2*miu(i)+3*n,length(t));
    for j=1:length(t)-100
        y(1,j)=yi(j+miu(i));
        for k=1:miu(i)
            if (j+miu(i)-k-1)<=0
                du(j+miu(i)-k)=ui1(j+miu(i)-k);
                du2(j+miu(i)-k)=ui2(j+miu(i)-k);
            else
                du(j+miu(i)-k)=ui1(j+miu(i)-k)-ui1(j+miu(i)-k-1);
                du2(j+miu(i)-k)=ui2(j+miu(i)-k)-ui2(j+miu(i)-k-1);
            end
            phi1(k,j)=du(j+miu(i)-k);
            phi1(k+miu(i),j)=du2(j+miu(i)-k);
        end
        for k=1:n
            if (j-k-1)<=0
                phi1(k+2*miu(i),j)=0;
            end
        end
    end
end

```

```

        phi1(k+2*miu(i)+n,j)=0;
    else
        phi1(k+2*miu(i),j)=ui1(j-k)-ui1(j-k-1);
        phi1(k+2*miu(i)+n,j)=ui2(j-k)-ui2(j-k-1);
    end
    if (j-k)<=0
        phi1(k+2*miu(i)+2*n,j)=0;
    else
        phi1(k+2*miu(i)+2*n,j)=yi(j-k);
    end
end
if i==1
    %theta1(:,j)=(phi1(:,j)*phi1(:,j))\'\'(phi1(:,j)*y(1,j)); % we could use this method (right inverse)
    theta1(:,j)=pinv(phi1(:,j))\'\'*y(1,j);
else
    %theta2(:,j)=(phi1(:,j)*phi1(:,j))\'\'(phi1(:,j)*y(1,j)); % we could use this method (right inverse)
    theta2(:,j)=pinv(phi1(:,j))\'\'*y(1,j);
end
end
end
%.....
G1i=zeros(2,2,length(t)-100); % these comments are related to the state which we use IIR for the future
G2i=zeros(2,2,length(t)-100);
M1_=zeros(2,n,length(t)-100);
M2_=zeros(2,n,length(t)-100);
F=zeros(2,n,length(t)-100);
for q=1:length(t)-100
    G1i(:,q)=[theta1(miu(1),q) theta1(miu(1)-m1,q); theta2(miu(2),q) theta2(miu(2)-m1,q)];
    G2i(:,q)=[theta1(2*miu(1),q) theta1(2*miu(1)-m1,q); theta2(2*miu(2),q) theta2(2*miu(2)-m1,q)];
    M1_(:,q)=[theta1(2*miu(1)+1:n+2*miu(1),q)'; theta2(2*miu(2)+1:n+2*miu(2),q)'];
    M2_(:,q)=[theta1(2*miu(1)+1+n:2*n+2*miu(1),q)'; theta2(2*miu(2)+n+1:2*n+2*miu(2),q)'];
    F(:,q)=[theta1(2*miu(1)+2*n+1:3*n+2*miu(1),q)'; theta2(2*miu(2)+2*n+1:2*miu(2)+3*n,q)'];
end
Gi=[G1i G2i];
%.....Toeplitz Matrix.....
b1 = zeros(1,P); b1(1,1)= g1(2);
a1 = g1(2:P+1);
G1 = toeplitz(a1,b1);
G1(:,M) = G1(:,M:P)*ones(P-M+1,1);
G1 = G1(:,1:M);
%.....
b2 = zeros(1,P); b2(1,1)= g2(2);
a2 = g2(2:P+1);
G2 = toeplitz(a2,b2);
G2(:,M) = G2(:,M:P)*ones(P-M+1,1);
G2 = G2(:,1:M);
%.....
G11=[G1(miu(1),1) G1(miu(1),m1+1); G1(miu(2),1) G1(miu(2),m1+1)];
G21=[G2(miu(1),1) G2(miu(1),m1+1); G2(miu(2),1) G2(miu(2),m1+1)];
G=[G11 G21];
M_=[M1_ M2_];
%.....
gamma = 1/60;
gain_DC=(num1(1)+num1(2)+num1(3))/(den1(1)+den1(2)+den1(3));
gain_DC2=(num2(1)+num2(2)+num2(3))/(den2(1)+den2(2)+den2(3));
Q = eye(2);

```

```

R1=((1.2)^2)*gamma*gain_DC^2*eye(2);
R2=gamma*gain_DC^2*eye(2);
R=[R1 zeros(2); zeros(2) R2];
alpha=0.5;
%for q=1:length(t)-100
    Kpfc=(G'*Q*G+R)\(G'*Q);
%end
%.....

x01=0.0882;
x02=441.2;
dU1_=zeros(n,length(t)-100);
dU2_=zeros(n,length(t)-100);
dU_=[dU1_;dU2_];
d1=zeros(1,length(t)-100);
y1=441.2;
u_1=[];
u_2=[];
ym=[];
y=0;
Y_d=zeros(2,length(t)-100);
Y_past=zeros(2,length(t)-100);
Y_m=zeros(2,length(t)-100);
D=zeros(2,length(t)-100);
E=zeros(2,length(t)-100);
dU1=zeros(2,length(t)-100);
dU2=zeros(2,length(t)-100);
dU=[dU1;dU2];
U1=zeros(M,length(t)-100);
U2=zeros(M,length(t)-100);
Y_=zeros(n,length(t)-100);
%.....step.....
r =ones(length(t)-100,1);
%.....
for i=1:length(t)-101

for j=1:2
    Y_d(j,i+1)=(alpha^j)*y+(1-(alpha)^j)*r(i+1); % Programmed
end

Y_past(:,i+1)=M_(:,i+1)*dU_(:,i+1)+F(:,i+1)*Y_(:,i+1);
D(:,i+1)=d1(i+1)*ones(2,1);

E(:,i+1)=Y_d(:,i+1)-Y_past(:,i+1)-D(:,i+1);

dU(:,i+1)=Kpfc*E(:,i+1);
dU1(:,i+1)=dU(1:2,i+1);
dU2(:,i+1)=dU(3:4,i+1);
U1(1,i+1)=dU1(1,i+1)+U1(1,i);
U2(1,i+1)=dU2(1,i+1)+U2(1,i);
dU(:,i+1)=[dU1(:,i+1);dU2(:,i+1)];

Y_m(:,i+1)=G*dU(:,i+1)+Y_past(:,i+1);

```

```

dU1_(2:n,i+2) = dU1_(1:n-1,i+1);
dU1_(1,i+2)=dU1(1,i+1);
dU2_(2:n,i+2) = dU2_(1:n-1,i+1);
dU2_(1,i+2)=dU2(1,i+1);
dU_(:,i+2)=[dU1_(:,i+2);dU2_(:,i+2)];
%Y_(2:n,i+2+miu(1))=Y_(1:n-1,i+1+miu(1)); % without using y(t+1) we can make Y_past in this way
% Y_(1,i+2+miu(1))=Y_m(1,i+1);

u1=U1(1,i+1);
u2=U2(1,i+1);
sim('Model');
d1(i+2)=y(end)-yl(end);
Y_(2:n,i+2)=Y_(1:n-1,i+1);
Y_(1,i+2)=yl(end);
y=y(end);%+dist(i,1); % nonlinear
y1=[y1; y+441.2];
ym=[ym; yl(end)];
u_1=[u_1; u1];
u_2=[u_2; u2];
%noise=[noise; n];
x01=x1(end);
x02=x2(end);
end
figure(3);
subplot(2,2,1:2);
hold on
plot(y1,'m');
hold on
plot(r+441.2,'r');
legend('Ts=0.05','Ts=0.1','Ts=0.5','r');
subplot(2,2,3);
hold on
plot(u_1,'m');
legend('Ts=0.05','Ts=0.1','Ts=0.5');
subplot(2,2,4);
hold on
plot(u_2,'m');
legend('Ts=0.05','Ts=0.1','Ts=0.5');

% veriyng mius.....
% .....
% 'a'
clear
clc
[n1,d1,n2,d2]=Inputsys(1);
Gs1 = tf(n1,d1);
Ts=0.1;
Gd1 = c2d(Gs1,Ts,'zoh');
[num1,den1]=tfdata(Gd1,'v');
Gs2 = tf(n2,d2);
Gd2 = c2d(Gs2,Ts,'zoh');
[num2,den2]=tfdata(Gd2,'v');
sys_info = stepinfo(Gd1);
ts1 = sys_info.SettlingTime;
tr1=sys_info.RiseTime;

```

```

sys_info = stepinfo(Gd2);
ts2 = sys_info.SettlingTime;
tr2=sys_info.RiseTime;
t=1:Ts:80;
[g1,t1] = step(Gd1,t);
[g2,t2] = step(Gd2,t);
P1=floor(tr1/Ts);
P2=floor(tr2/Ts);
N1=floor( ts1/Ts);
N2=floor( ts2/Ts);
%P=max(P1,P2);
P=7;
N=max(N1,N2);
M=P;
n=3;
m1=2;
miu=[3 5];
ui1=randn(1,length(t));
ui2=randn(1,length(t));
ui=[ui1; ui2];
Gd=[Gd1 Gd2];
[yi,t1]=lsim(Gd,ui,t);
theta1=zeros(2*miu(1)+3*n,length(t)-100);
theta2=zeros(2*miu(2)+3*n,length(t)-100);
for i=1:2
    phi1=zeros(2*miu(i)+3*n,length(t));
    for j=1:length(t)-100
        y(1,j)=yi(j+miu(i));
        for k=1:miu(i)
            if (j+miu(i)-k-1)<=0
                du(j+miu(i)-k)=ui1(j+miu(i)-k);
                du2(j+miu(i)-k)=ui2(j+miu(i)-k);
            else
                du(j+miu(i)-k)=ui1(j+miu(i)-k)-ui1(j+miu(i)-k-1);
                du2(j+miu(i)-k)=ui2(j+miu(i)-k)-ui2(j+miu(i)-k-1);
            end
            phi1(k,j)=du(j+miu(i)-k);
            phi1(k+miu(i),j)=du2(j+miu(i)-k);
        end
        for k=1:n
            if (j-k-1)<=0
                phi1(k+2*miu(i),j)=0;
                phi1(k+2*miu(i)+n,j)=0;
            else
                phi1(k+2*miu(i),j)=ui1(j-k)-ui1(j-k-1);
                phi1(k+2*miu(i)+n,j)=ui2(j-k)-ui2(j-k-1);
            end
            if (j-k)<=0
                phi1(k+2*miu(i)+2*n,j)=0;
            else
                phi1(k+2*miu(i)+2*n,j)=yi(j-k);
            end
        end
    end
    if i==1
        %theta1(:,j)=(phi1(:,j)*phi1(:,j))\'(phi1(:,j)*y(1,j)); % we could use this method (right inverse)
        theta1(:,j)=pinv(phi1(:,j))\'*y(1,j);
    end
end

```

```

else
    %theta2(:,j)=(phi1(:,j)*phi1(:,j))\'(phi1(:,j)*y(1,j)); % we could use this method (right inverse)
    theta2(:,j)=pinv(phi1(:,j))\'*y(1,j);
end
end
end
%.....
G1i=zeros(2,2,length(t)-100); % these comments are related to the state which we use IIR for the future
G2i=zeros(2,2,length(t)-100);
M1_=zeros(2,n,length(t)-100);
M2_=zeros(2,n,length(t)-100);
F=zeros(2,n,length(t)-100);
for q=1:length(t)-100
    G1i(:,q)=[theta1(miu(1),q) theta1(miu(1)-m1,q); theta2(miu(2),q) theta2(miu(2)-m1,q)];
    G2i(:,q)=[theta1(2*miu(1),q) theta1(2*miu(1)-m1,q); theta2(2*miu(2),q) theta2(2*miu(2)-m1,q)];
    M1_(:,q)=[theta1(2*miu(1)+1:n+2*miu(1),q)'; theta2(2*miu(2)+1:n+2*miu(2),q)'];
    M2_(:,q)=[theta1(2*miu(1)+1+n:2*n+2*miu(1),q)'; theta2(2*miu(2)+n+1:2*n+2*miu(2),q)'];
    F(:,q)=[theta1(2*miu(1)+2*n+1:3*n+2*miu(1),q)'; theta2(2*miu(2)+2*n+1:2*miu(2)+3*n,q)'];
end
Gi=[G1i G2i];
%.....Toeplitz Matrix.....
b1 = zeros(1,P); b1(1,1)= g1(2);
a1 = g1(2:P+1);
G1 = toeplitz(a1,b1);
G1(:,M) = G1(:,M:P)*ones(P-M+1,1);
G1 = G1(:,1:M);
%.....
b2 = zeros(1,P); b2(1,1)= g2(2);
a2 = g2(2:P+1);
G2 = toeplitz(a2,b2);
G2(:,M) = G2(:,M:P)*ones(P-M+1,1);
G2 = G2(:,1:M);
%.....
G11=[G1(miu(1),1) G1(miu(1),m1+1); G1(miu(2),1) G1(miu(2),m1+1)];
G21=[G2(miu(1),1) G2(miu(1),m1+1); G2(miu(2),1) G2(miu(2),m1+1)];
G=[G11 G21];
M_=[M1_ M2_];
%.....
gamma =1/60;
gain_DC=(num1(1)+num1(2)+num1(3))/(den1(1)+den1(2)+den1(3));
gain_DC2=(num2(1)+num2(2)+num2(3))/(den2(1)+den2(2)+den2(3));
Q = eye(2);
R1=((1.2)^2)*gamma*gain_DC^2*eye(2);
R2=gamma*gain_DC2^2*eye(2);
R=[R1 zeros(2); zeros(2) R2];
alpha=0.5;
%for q=1:length(t)-100
    Kpfc=(G'*Q*G+R)\(G'*Q);
%end
%.....

x01=0.0882;
x02=441.2;
dU1_=zeros(n,length(t)-100);
dU2_=zeros(n,length(t)-100);
dU_=[dU1_;dU2_];

```



```

d1=zeros(1,length(t)-100);
y1=441.2;
u_1=[];
u_2=[];
ym=[];
y=0;
Y_d=zeros(2,length(t)-100);
Y_past=zeros(2,length(t)-100);
Y_m=zeros(2,length(t)-100);
D=zeros(2,length(t)-100);
E=zeros(2,length(t)-100);
dU1=zeros(2,length(t)-100);
dU2=zeros(2,length(t)-100);
dU=[dU1;dU2];
U1=zeros(M,length(t)-100);
U2=zeros(M,length(t)-100);
Y_=zeros(n,length(t)-100);
% .....step.....
r =ones(length(t)-100,1);
% .....
for i=1:length(t)-101

for j=1:2
    Y_d(j,i+1)=(alpha^j)*y+(1-(alpha)^j)*r(i+1); % Programmed
end

Y_past(:,i+1)=M(:,i+1)*dU(:,i+1)+F(:,i+1)*Y_(:,i+1);
D(:,i+1)=d1(i+1)*ones(2,1);

E(:,i+1)=Y_d(:,i+1)-Y_past(:,i+1)-D(:,i+1);

dU(:,i+1)=Kpfc*E(:,i+1);
dU1(:,i+1)=dU(1:2,i+1);
dU2(:,i+1)=dU(3:4,i+1);
U1(1,i+1)=dU1(1,i+1)+U1(1,i);
U2(1,i+1)=dU2(1,i+1)+U2(1,i);
dU(:,i+1)=[dU1(:,i+1);dU2(:,i+1)];

Y_m(:,i+1)=G*dU(:,i+1)+Y_past(:,i+1);

dU1_(2:n,i+2) = dU1_(1:n-1,i+1);
dU1_(1,i+2)=dU1(1,i+1);
dU2_(2:n,i+2) = dU2_(1:n-1,i+1);
dU2_(1,i+2)=dU2(1,i+1);
dU_(:,i+2)=[dU1_(:,i+2);dU2_(:,i+2)];
% Y_(2:n,i+2+miu(1))=Y_(1:n-1,i+1+miu(1)); % without using y(t+1) we can make Y_past in this way
% Y_(1,i+2+miu(1))=Y_m(1,i+1);

u1=U1(1,i+1);
u2=U2(1,i+1);
sim('Model');
d1(i+2)=y(end)-y1(end);
Y_(2:n,i+2)=Y_(1:n-1,i+1);
Y_(1,i+2)=y1(end);

```

```

y=y(end);%+dist(i,1); % nonlinear
y1=[y1; y+441.2];
ym=[ym; yl(end)];
u_1=[u_1; u1];
u_2=[u_2; u2];
%noise=[noise; n];
x01=x1(end);
x02=x2(end);
end
figure(3);
subplot(2,2,1:2);
plot(y1,'b');
grid on
title('Response of the nonlinear system');
xlabel('sample');
subplot(2,2,3);
plot(u_1,'b');
grid on
xlabel('sample');
title('Control law for input 1 without bias');
subplot(2,2,4);
plot(u_2,'b');
grid on
xlabel('sample');
title('Control law for input 2 without bias');
% .....
% 'b'
clear
clc
[n1,d1,n2,d2]=Inputsys(1);
Gs1 = tf(n1,d1);
Ts=0.1;
Gd1 = c2d(Gs1,Ts,'zoh');
[num1,den1]=tfdata(Gd1,'v');
Gs2 = tf(n2,d2);
Gd2 = c2d(Gs2,Ts,'zoh');
[num2,den2]=tfdata(Gd2,'v');
sys_info = stepinfo(Gd1);
ts1 = sys_info.SettlingTime;
tr1=sys_info.RiseTime;
sys_info = stepinfo(Gd2);
ts2 = sys_info.SettlingTime;
tr2=sys_info.RiseTime;
t=1:Ts:80;
[g1,t1] = step(Gd1,t);
[g2,t2] = step(Gd2,t);
P1=floor(tr1/Ts);
P2=floor(tr2/Ts);
N1=floor( ts1/Ts);
N2=floor( ts2/Ts);
%P=max(P1,P2);
P=7;
N=max(N1,N2);
M=P;
n=3;
m1=2;

```

```

miu=[3 6];
ui1=randn(1,length(t));
ui2=randn(1,length(t));
ui=[ui1; ui2];
Gd=[Gd1 Gd2];
[yi,t1]=lsim(Gd,ui,t);
theta1=zeros(2*miu(1)+3*n,length(t)-100);
theta2=zeros(2*miu(2)+3*n,length(t)-100);
for i=1:2
    phi1=zeros(2*miu(i)+3*n,length(t));
    for j=1:length(t)-100
        y(1,j)=yi(j+miu(i));
        for k=1:miu(i)
            if (j+miu(i)-k-1)<=0
                du(j+miu(i)-k)=ui1(j+miu(i)-k);
                du2(j+miu(i)-k)=ui2(j+miu(i)-k);
            else
                du(j+miu(i)-k)=ui1(j+miu(i)-k)-ui1(j+miu(i)-k-1);
                du2(j+miu(i)-k)=ui2(j+miu(i)-k)-ui2(j+miu(i)-k-1);
            end
            phi1(k,j)=du(j+miu(i)-k);
            phi1(k+miu(i),j)=du2(j+miu(i)-k);
        end
        for k=1:n
            if (j-k-1)<=0
                phi1(k+2*miu(i),j)=0;
                phi1(k+2*miu(i)+n,j)=0;
            else
                phi1(k+2*miu(i),j)=ui1(j-k)-ui1(j-k-1);
                phi1(k+2*miu(i)+n,j)=ui2(j-k)-ui2(j-k-1);
            end
            if (j-k)<=0
                phi1(k+2*miu(i)+2*n,j)=0;
            else
                phi1(k+2*miu(i)+2*n,j)=yi(j-k);
            end
        end
    end
    if i==1
        %theta1(:,j)=(phi1(:,j)*phi1(:,j))\'(phi1(:,j)*y(1,j)); % we could use this method (right inverse)
        theta1(:,j)=pinv(phi1(:,j))\'*y(1,j);
    else
        %theta2(:,j)=(phi1(:,j)*phi1(:,j))\'(phi1(:,j)*y(1,j)); % we could use this method (right inverse)
        theta2(:,j)=pinv(phi1(:,j))\'*y(1,j);
    end
end
end
%.....
G1i=zeros(2,2,length(t)-100); % these comments are related to the state which we use IIR for the future
G2i=zeros(2,2,length(t)-100);
M1_=zeros(2,n,length(t)-100);
M2_=zeros(2,n,length(t)-100);
F=zeros(2,n,length(t)-100);
for q=1:length(t)-100
    G1i(:, :, q)=[theta1(miu(1),q) theta1(miu(1)-m1,q); theta2(miu(2),q) theta2(miu(2)-m1,q)];
    G2i(:, :, q)=[theta1(2*miu(1),q) theta1(2*miu(1)-m1,q); theta2(2*miu(2),q) theta2(2*miu(2)-m1,q)];
    M1_(:, :, q)=[theta1(2*miu(1)+1:n+2*miu(1),q)'; theta2(2*miu(2)+1:n+2*miu(2),q)'];

```

```

M2_(:,q)=[theta1(2*miu(1)+1+n:2*n+2*miu(1),q)'; theta2(2*miu(2)+n+1:2*n+2*miu(2),q)'];
F(:,q)=[theta1(2*miu(1)+2*n+1:3*n+2*miu(1),q)'; theta2(2*miu(2)+2*n+1:2*miu(2)+3*n,q)'];
end
Gi=[G1i G2i];
%.....Toeplitz Matrix.....
b1 = zeros(1,P); b1(1,1)= g1(2);
a1 = g1(2:P+1);
G1 = toeplitz(a1,b1);
G1(:,M) = G1(:,M:P)*ones(P-M+1,1);
G1 = G1(:,1:M);
%.....
b2 = zeros(1,P); b2(1,1)= g2(2);
a2 = g2(2:P+1);
G2 = toeplitz(a2,b2);
G2(:,M) = G2(:,M:P)*ones(P-M+1,1);
G2 = G2(:,1:M);
%.....
G11=[G1(miu(1),1) G1(miu(1),m1+1); G1(miu(2),1) G1(miu(2),m1+1)];
G21=[G2(miu(1),1) G2(miu(1),m1+1); G2(miu(2),1) G2(miu(2),m1+1)];
G=[G11 G21];
M_=[M1_ M2_];
%.....
gamma =1/60;
gain_DC=(num1(1)+num1(2)+num1(3))/(den1(1)+den1(2)+den1(3));
gain_DC2=(num2(1)+num2(2)+num2(3))/(den2(1)+den2(2)+den2(3));
Q = eye(2);
R1=((1.2)^2)*gamma*gain_DC^2*eye(2);
R2=gamma*gain_DC2^2*eye(2);
R=[R1 zeros(2); zeros(2) R2];
alpha=0.5;
%for q=1:length(t)-100
    Kpfc=(G'*Q*G+R)\(G'*Q);
%end
%.....

x01=0.0882;
x02=441.2;
dU1_=zeros(n,length(t)-100);
dU2_=zeros(n,length(t)-100);
dU_=[dU1_;dU2_];
d1=zeros(1,length(t)-100);
y1=441.2;
u_1=[];
u_2=[];
ym=[];
y=0;
Y_d=zeros(2,length(t)-100);
Y_past=zeros(2,length(t)-100);
Y_m=zeros(2,length(t)-100);
D=zeros(2,length(t)-100);
E=zeros(2,length(t)-100);
dU1=zeros(2,length(t)-100);
dU2=zeros(2,length(t)-100);
dU=[dU1;dU2];
U1=zeros(M,length(t)-100);
U2=zeros(M,length(t)-100);

```

```

Y_=zeros(n,length(t)-100);
%.....step.....
r=ones(length(t)-100,1);
%.....
for i=1:length(t)-101

for j=1:2
    Y_d(j,i+1)=(alpha^j)*y+(1-(alpha)^j)*r(i+1); % Programmed
end

Y_past(:,i+1)=M_(:,i+1)*dU_(:,i+1)+F(:,i+1)*Y_(:,i+1);
D(:,i+1)=d1(i+1)*ones(2,1);

E(:,i+1)=Y_d(:,i+1)-Y_past(:,i+1)-D(:,i+1);

dU(:,i+1)=Kpfc*E(:,i+1);
dU1(:,i+1)=dU(1:2,i+1);
dU2(:,i+1)=dU(3:4,i+1);
U1(1,i+1)=dU1(1,i+1)+U1(1,i);
U2(1,i+1)=dU2(1,i+1)+U2(1,i);
dU(:,i+1)=[dU1(:,i+1);dU2(:,i+1)];

Y_m(:,i+1)=G*dU(:,i+1)+Y_past(:,i+1);

dU1_(2:n,i+2)=dU1_(1:n-1,i+1);
dU1_(1,i+2)=dU1(1,i+1);
dU2_(2:n,i+2)=dU2_(1:n-1,i+1);
dU2_(1,i+2)=dU2(1,i+1);
dU_(:,i+2)=[dU1_(:,i+2);dU2_(:,i+2)];
%Y_(2:n,i+2+miu(1))=Y_(1:n-1,i+1+miu(1)); % without using y(t+1) we can make Y_past in this way
% Y_(1,i+2+miu(1))=Y_m(1,i+1);

u1=U1(1,i+1);
u2=U2(1,i+1);
sim('Model');
d1(i+2)=y(end)-yl(end);
Y_(2:n,i+2)=Y_(1:n-1,i+1);
Y_(1,i+2)=yl(end);
y=y(end);%+dist(i,1); % nonlinear
y1=[y1; y+441.2];
ym=[ym; yl(end)];
u_1=[u_1; u1];
u_2=[u_2; u2];
%noise=[noise; n];
x01=x1(end);
x02=x2(end);
end
figure(3);
subplot(2,2,1:2);
hold on
plot(y1,'c');
subplot(2,2,3);
hold on
plot(u_1,'c');

```

```

subplot(2,2,4);
hold on
plot(u_2,'c');
%.....
% 'c'
clear
clc
[n1,d1,n2,d2]=Inputsys(1);
Gs1 = tf(n1,d1);
Ts=0.1;
Gd1 = c2d(Gs1,Ts,'zoh');
[num1,den1]=tfdata(Gd1,'v');
Gs2 = tf(n2,d2);
Gd2 = c2d(Gs2,Ts,'zoh');
[num2,den2]=tfdata(Gd2,'v');
sys_info = stepinfo(Gd1);
ts1 = sys_info.SettlingTime;
tr1=sys_info.RiseTime;
sys_info = stepinfo(Gd2);
ts2 = sys_info.SettlingTime;
tr2=sys_info.RiseTime;
t=1:Ts:80;
[g1,t1] = step(Gd1,t);
[g2,t2] = step(Gd2,t);
P1=floor(tr1/Ts);
P2=floor(tr2/Ts);
N1=floor( ts1/Ts);
N2=floor( ts2/Ts);
%P=max(P1,P2);
P=7;
N=max(N1,N2);
M=P;
n=3;
m1=2;
miu=[3 7];
ui1=randn(1,length(t));
ui2=randn(1,length(t));
ui=[ui1; ui2];
Gd=[Gd1 Gd2];
[yi,t1]=lsim(Gd,ui,t);
theta1=zeros(2*miu(1)+3*n,length(t)-100);
theta2=zeros(2*miu(2)+3*n,length(t)-100);
for i=1:2
    phi1=zeros(2*miu(i)+3*n,length(t));
    for j=1:length(t)-100
        y(1,j)=yi(j+miu(i));
        for k=1:miu(i)
            if (j+miu(i)-k-1)<=0
                du(j+miu(i)-k)=ui1(j+miu(i)-k);
                du2(j+miu(i)-k)=ui2(j+miu(i)-k);
            else
                du(j+miu(i)-k)=ui1(j+miu(i)-k)-ui1(j+miu(i)-k-1);
                du2(j+miu(i)-k)=ui2(j+miu(i)-k)-ui2(j+miu(i)-k-1);
            end
            phi1(k,j)=du(j+miu(i)-k);
            phi1(k+miu(i),j)=du2(j+miu(i)-k);
        end
    end
end

```

```

end
for k=1:n
    if (j-k-1)<=0
        phi1(k+2*miu(i),j)=0;
        phi1(k+2*miu(i)+n,j)=0;
    else
        phi1(k+2*miu(i),j)=ui1(j-k)-ui1(j-k-1);
        phi1(k+2*miu(i)+n,j)=ui2(j-k)-ui2(j-k-1);
    end
    if (j-k)<=0
        phi1(k+2*miu(i)+2*n,j)=0;
    else
        phi1(k+2*miu(i)+2*n,j)=yi(j-k);
    end
end
if i==1
    %theta1(:,j)=(phi1(:,j)*phi1(:,j))\ (phi1(:,j)*y(1,j)); % we could use this method (right inverse)
    theta1(:,j)=pinv(phi1(:,j))*y(1,j);
else
    %theta2(:,j)=(phi1(:,j)*phi1(:,j))\ (phi1(:,j)*y(1,j)); % we could use this method (right inverse)
    theta2(:,j)=pinv(phi1(:,j))*y(1,j);
end
end
end
% .....
G1i=zeros(2,2,length(t)-100); % these comments are related to the state which we use IIR for the future
G2i=zeros(2,2,length(t)-100);
M1_=zeros(2,n,length(t)-100);
M2_=zeros(2,n,length(t)-100);
F=zeros(2,n,length(t)-100);
for q=1:length(t)-100
    G1i(:,q)=[theta1(miu(1),q) theta1(miu(1)-m1,q); theta2(miu(2),q) theta2(miu(2)-m1,q)];
    G2i(:,q)=[theta1(2*miu(1),q) theta1(2*miu(1)-m1,q); theta2(2*miu(2),q) theta2(2*miu(2)-m1,q)];
    M1_(:,q)=[theta1(2*miu(1)+1:n+2*miu(1),q)'; theta2(2*miu(2)+1:n+2*miu(2),q)'];
    M2_(:,q)=[theta1(2*miu(1)+1+n:2*n+2*miu(1),q)'; theta2(2*miu(2)+n+1:2*n+2*miu(2),q)'];
    F(:,q)=[theta1(2*miu(1)+2*n+1:3*n+2*miu(1),q)'; theta2(2*miu(2)+2*n+1:2*miu(2)+3*n,q)'];
end
Gi=[G1i G2i];
% .....Toeplitz Matrix.....
b1 = zeros(1,P); b1(1,1)= g1(2);
a1 = g1(2:P+1);
G1 = toeplitz(a1,b1);
G1(:,M) = G1(:,M:P)*ones(P-M+1,1);
G1 = G1(:,1:M);
% .....
b2 = zeros(1,P); b2(1,1)= g2(2);
a2 = g2(2:P+1);
G2 = toeplitz(a2,b2);
G2(:,M) = G2(:,M:P)*ones(P-M+1,1);
G2 = G2(:,1:M);
% .....
G11=[G1(miu(1),1) G1(miu(1),m1+1); G1(miu(2),1) G1(miu(2),m1+1)];
G21=[G2(miu(1),1) G2(miu(1),m1+1); G2(miu(2),1) G2(miu(2),m1+1)];
G=[G11 G21];
M_=[M1_ M2_];
% .....

```

```

gamma =1/60;
gain_DC=(num1(1)+num1(2)+num1(3))/(den1(1)+den1(2)+den1(3));
gain_DC2=(num2(1)+num2(2)+num2(3))/(den2(1)+den2(2)+den2(3));
Q = eye(2);
R1 =((1.2)^2)*gamma*gain_DC^2*eye(2);
R2=gamma*gain_DC2^2*eye(2);
R=[R1 zeros(2); zeros(2) R2];
alpha=0.5;
%for q=1:length(t)-100
    Kpfc=(G'*Q*G+R)\(G'*Q);
%end
%.....

```

```

x01=0.0882;
x02=441.2;
dU1_=zeros(n,length(t)-100);
dU2_=zeros(n,length(t)-100);
dU_=[dU1_;dU2_];
d1=zeros(1,length(t)-100);
y1=441.2;
u_1=[];
u_2=[];
ym=[];
y=0;
Y_d=zeros(2,length(t)-100);
Y_past=zeros(2,length(t)-100);
Y_m=zeros(2,length(t)-100);
D=zeros(2,length(t)-100);
E=zeros(2,length(t)-100);
dU1=zeros(2,length(t)-100);
dU2=zeros(2,length(t)-100);
dU=[dU1;dU2];
U1=zeros(M,length(t)-100);
U2=zeros(M,length(t)-100);
Y_=zeros(n,length(t)-100);
%.....step.....
r =ones(length(t)-100,1);
%.....
for i=1:length(t)-101

for j=1:2
    Y_d(j,i+1)=(alpha^j)*y+(1-(alpha)^j)*r(i+1); % Programmed
end

Y_past(:,i+1)=M_(:,i+1)*dU_(:,i+1)+F(:,i+1)*Y_(:,i+1);
D(:,i+1)=d1(i+1)*ones(2,1);

E(:,i+1)=Y_d(:,i+1)-Y_past(:,i+1)-D(:,i+1);

dU(:,i+1)=Kpfc*E(:,i+1);
dU1(:,i+1)=dU(1:2,i+1);
dU2(:,i+1)=dU(3:4,i+1);
U1(1,i+1)=dU1(1,i+1)+U1(1,i);
U2(1,i+1)=dU2(1,i+1)+U2(1,i);
dU(:,i+1)=[dU1(:,i+1);dU2(:,i+1)];

```



```

Y_m(:,i+1)=G*dU(:,i+1)+Y_past(:,i+1);

dU1_(2:n,i+2) = dU1_(1:n-1,i+1);
dU1_(1,i+2)=dU1(1,i+1);
dU2_(2:n,i+2) = dU2_(1:n-1,i+1);
dU2_(1,i+2)=dU2(1,i+1);
dU_(:,i+2)=[dU1_(:,i+2);dU2_(:,i+2)];
%Y_(2:n,i+2+miu(1))=Y_(1:n-1,i+1+miu(1)); % without using y(t+1) we can make Y_past in this way
% Y_(1,i+2+miu(1))=Y_m(1,i+1);

u1=U1(1,i+1);
u2=U2(1,i+1);
sim('Model');
d1(i+2)=y(end)-yl(end);
Y_(2:n,i+2)=Y_(1:n-1,i+1);
Y_(1,i+2)=yl(end);
y=y(end);%+dist(i,1); % nonlinear
y1=[y1; y+441.2];
ym=[ym; yl(end)];
u_1=[u_1; u1];
u_2=[u_2; u2];
%noise=[noise; n];
x01=x1(end);
x02=x2(end);
end
figure(3);
subplot(2,2,1:2);
hold on
plot(y1,'m');
hold on
plot(r+441.2,'r');
legend('a','b','c','r');
subplot(2,2,3);
hold on
plot(u_1,'m');
legend('a','b','c');
subplot(2,2,4);
hold on
plot(u_2,'m');
legend('a','b','c');

```