



پروژه سوم درس MPC

طراحی کنترل کننده GPC

نام دانشجو: فاطمه زاهدی

شماره دانشجویی: 94201583

نام استاد: دکتر حائری

فهرست:

1. معرفی سیستم مورد استفاده: 2
2. طراحی کنترل کننده GPC: 2
3. پاسخ سیستم به ازای خروجی مطلوب های مختلف: 3
 - 3.1. خروجی مطلوب پالس: 3
 - 3.2. خروجی مطلوب سینوس: 4
 - 3.3. خروجی مطلوب پله ای با پرش های مختلف: 5
 - 3.4. خروجی مطلوب پله ای: 5
4. بررسی اثر پارامترها: 7
 - 4.1. بررسی ماتریس R: 7
 - 4.2. بررسی ماتریس Q: 8
 - 4.3. بررسی پارامتر آلفا: 9
 - 4.4. بررسی پارامتر P یا M: 10
 - 4.5. بررسی پارامتر T_s : 10
5. بررسی اثر اغتشاش و نویز: 11
 - 5.1. بررسی اثر اغتشاش: 11
 - 5.2. بررسی اثر نویز: 12
6. بررسی عدم قطعیت در پروسه: 13
7. کدها و شبیه سازی ها: 14

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 GPC:

در این طراحی ابتدا بایستی به تعیین پارامترهای $T_s, P, N1, N2, d, n_b, \tilde{n}_a$ و M بپردازیم. مقدار T_s را مطابق با روشی که در پروژه قبل گفته شد بدست می آوریم که این مقدار برابر با 0.1 قرار داده شده است. در این طراحی مقدار M و P را برابر با هم قرار می دهیم. که این مقدار با توجه به رابطه $M = P = \frac{t_r}{T_s}$ بدست آورده ایم. با توجه به اینکه سیستم ما MIMO است ما مقدار بزرگتر را به عنوان M در نظر می گیریم. بنابراین در اینجا ما این مقدار را برابر با 5 گرفته ایم. سایر پارامترها چون d, n_b و \tilde{n}_a با توجه به سیستم بدست می آید و مقدار این مقادیر به ترتیب 0، 1 و 3 می باشد. مقادیر $N1$ و $N2$ نیز با استفاده از رابطه $N1=d+1$ و $N2=P+d$ بدست آمد که مقدار آنها به ترتیب 1 و 5 شد.

همانطور که میدانیم این کنترل کننده با استفاده از مدل IIR و FIR طراحی می شود بدین صورت که آینده از FIR و گذشته از IIR تشکیل می شود. قسمتی که مربوط به آینده است را می توان یا با استفاده از پاسخ پله یا با استفاده از پاسخ ضربه مدل FIR آن را تشکیل داد. که ما برای پیاده سازی این کنترل کننده از پاسخ پله استفاده کرده ایم. 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 یک و مقدار آلفا برابر با نیم در نظر گرفته شده است. همانطور که در این شکل می توان دید به دلیل استفاده از پاسخ پله برای قسمت FIR خطای حالت دائم پاسخ سیستم صفر می باشد.

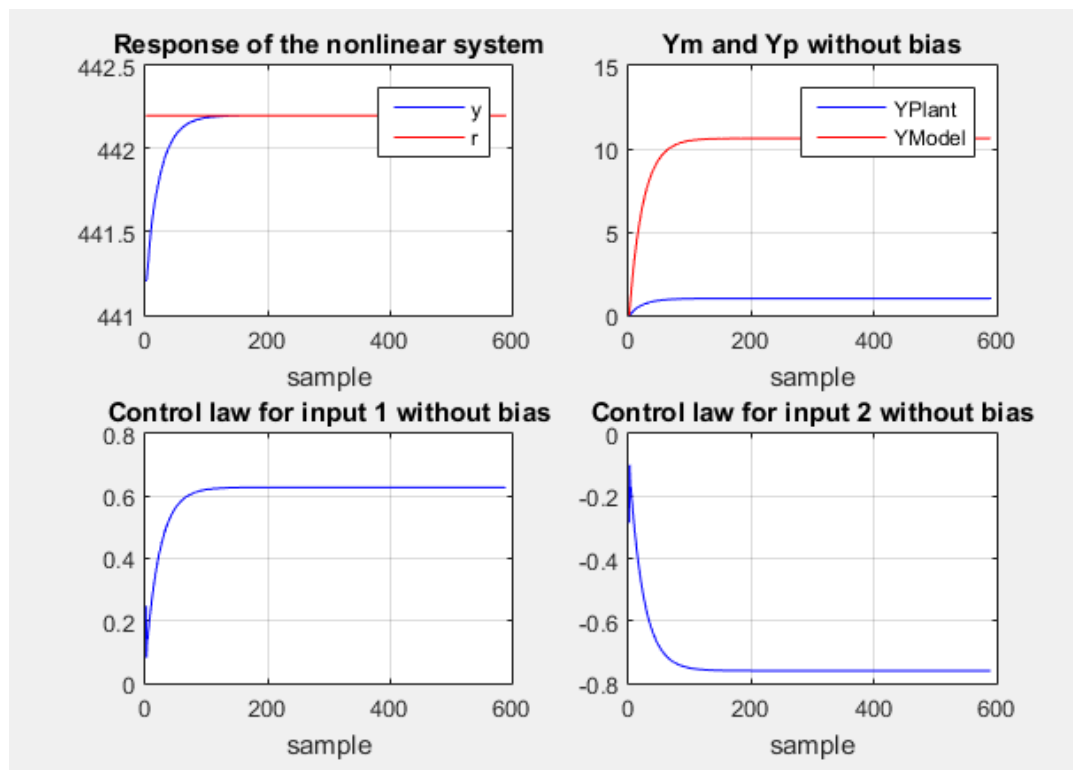


Figure1. GPC on nonlinear system.

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 یک 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.

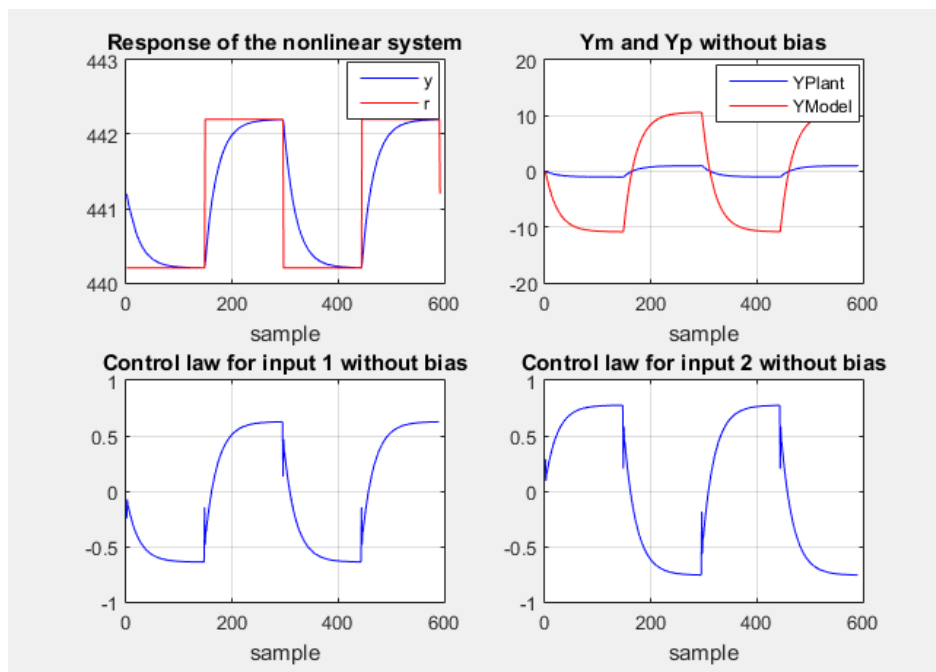


Figure 2. Response of the system with pulse as a desired output.

3.2. خروجی مطلوب سینوس:

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.

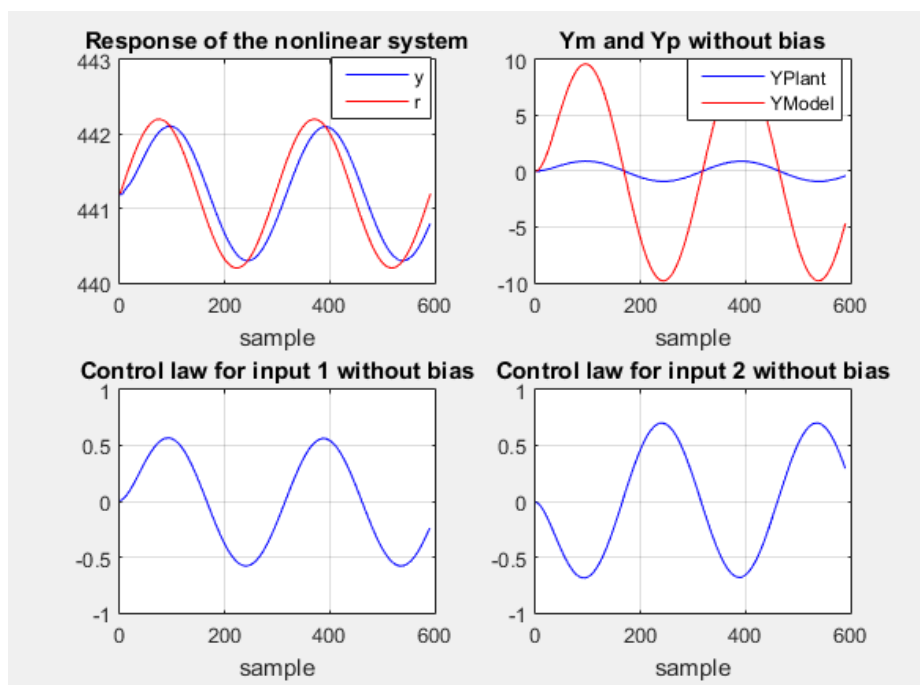


Figure 3. Response of the system with sine as a desired output.

همانطور که در شکل بالا می‌بینیم خروجی 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.

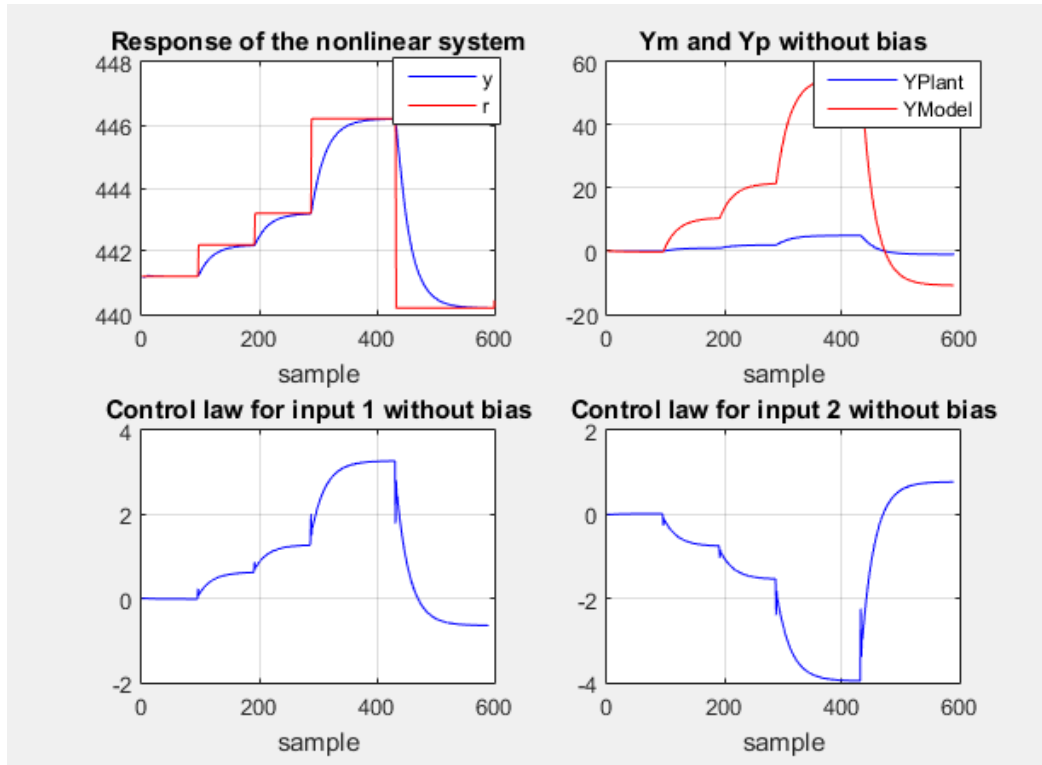


Figure 4. Response of the system with step with various jumps as a desired output.

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 0.25%، 1% و 3% نقطه کار رسم می‌کنیم. شکل زیر پاسخ را به ازای این خروجی مطلوب با دامنه 0.25% 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.

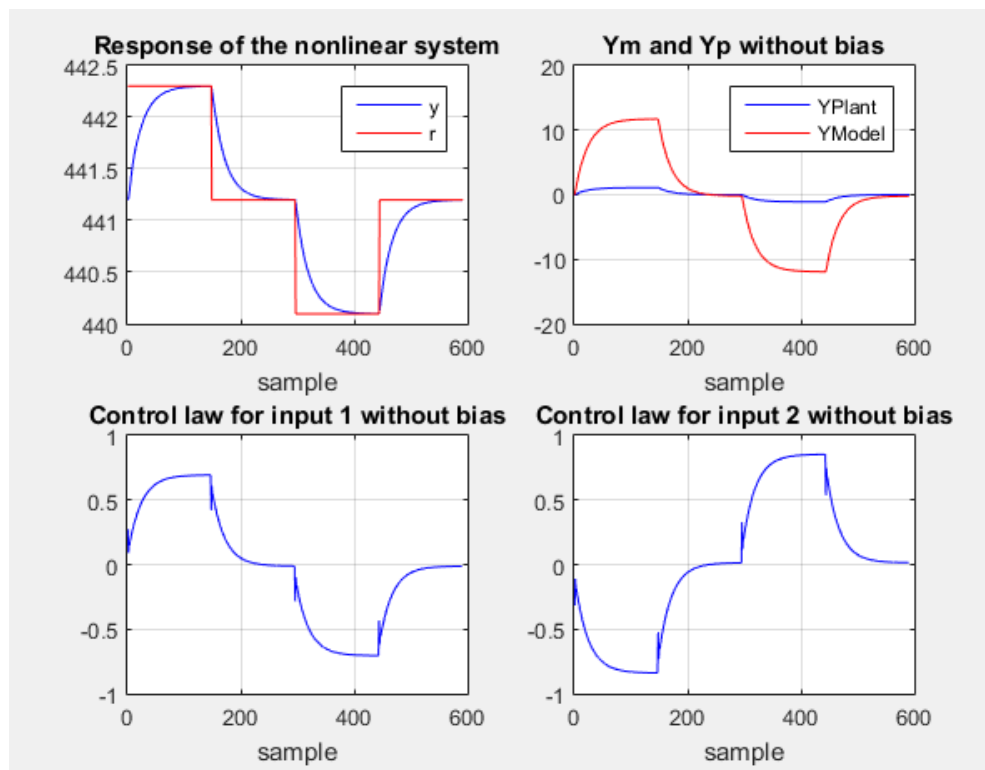


Figure 5. Response of the system with step with 0.25% of operating point as a desired output.

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

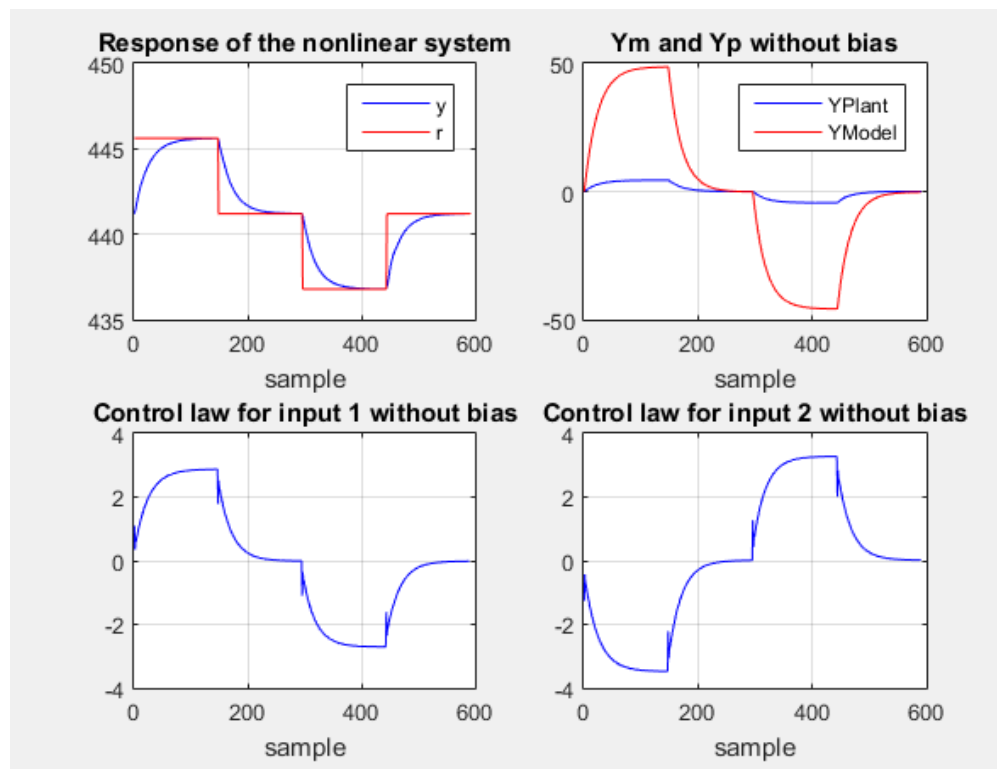


Figure 6. Response of the system with step with 1% of operating point as a desired output.

همانطور که در شکل بالا می‌بینیم پاسخ سیستم، خروجی مطلوب را دنبال 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 3% نقطه کار می‌بینیم.

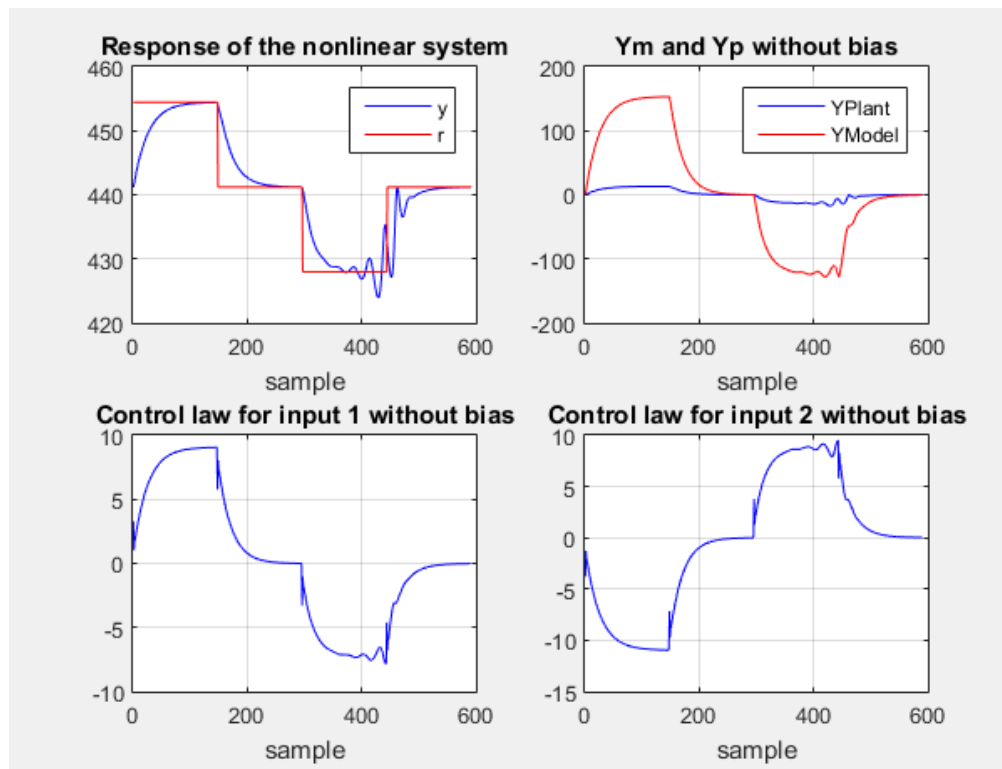


Figure 7. Response of the system with step with 3% of operating point as a desired output.

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 GPC 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، 1/2 و 1/600 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.

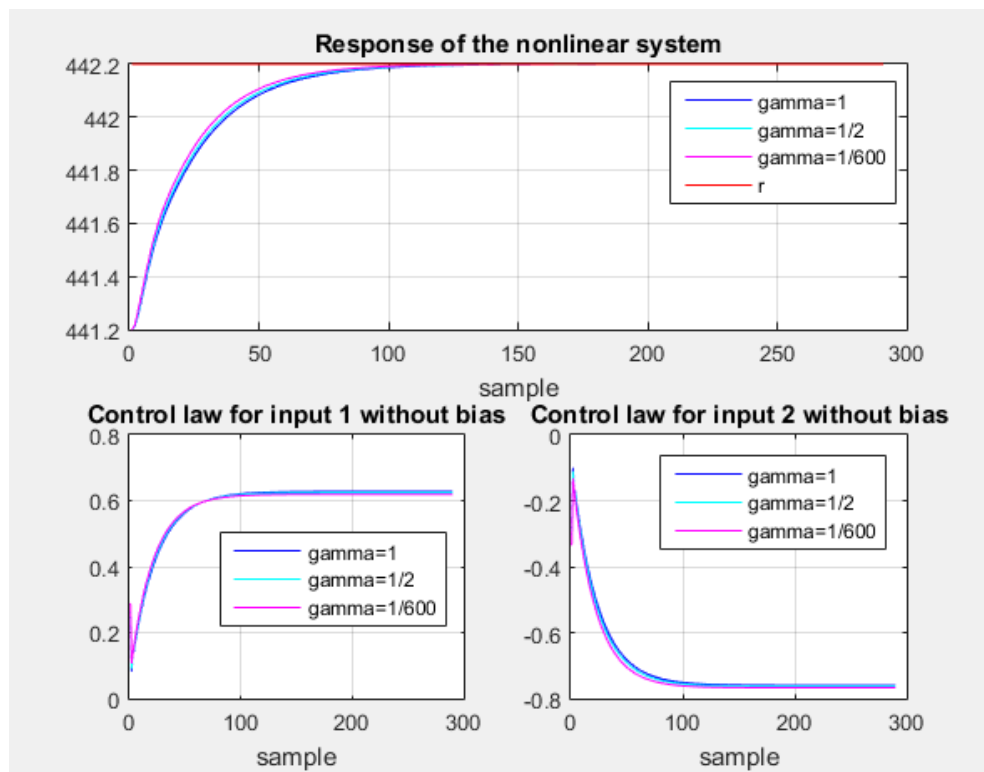


Figure 8. Responses of the system in verifying R .

همانطور که در شکل 8 می بینیم کاهش گاما تاثیری در حالت ماندگار پاسخ سیستم نمی گذارد و خطای پاسخ در حالت ماندگار همواره صفر است. اما همانطور که در این شکل می بینیم 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 $Q=I$ ، $Q=2I$ و $Q=60I$ بررسی می کنیم.

شکل 9 تاثیر افزایش Q را به ازای مقادیر گفته شده نشان می دهد.

همانطور که در این شکل می توان دید با افزایش Q پاسخ سیستم تندتر می شود. اما دقیقا مشابه کاهش گاما، خطای حالت ماندگار تغییری نمی کند و همواره صفر می ماند.

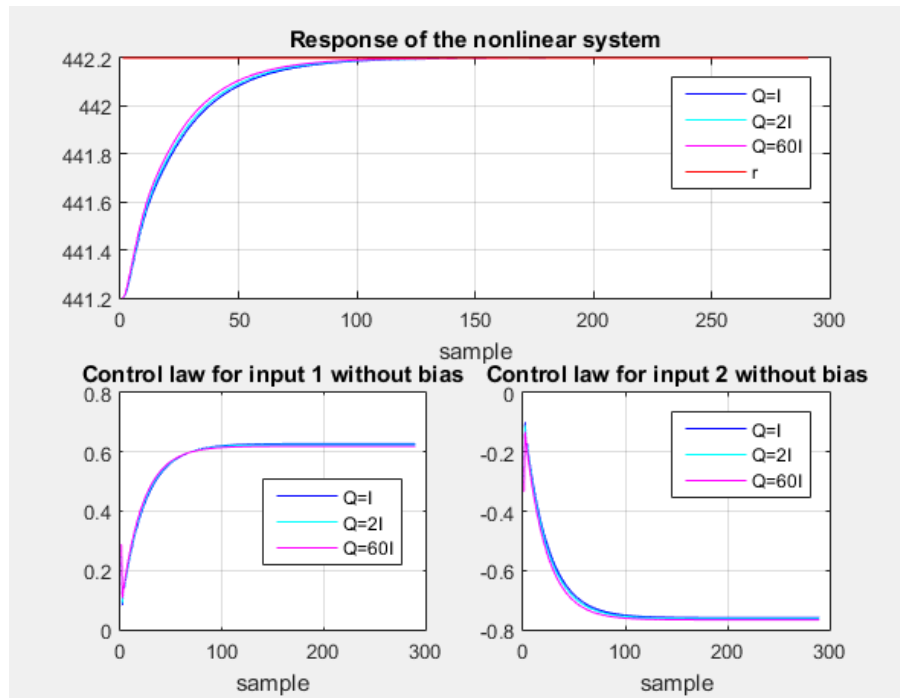


Figure 9. Responses of the system in verifying Q .

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 0.1، 0.5 و 0.8 نشان می دهد. در طی این بررسی مقدار گاما یک قرار داده شده است.

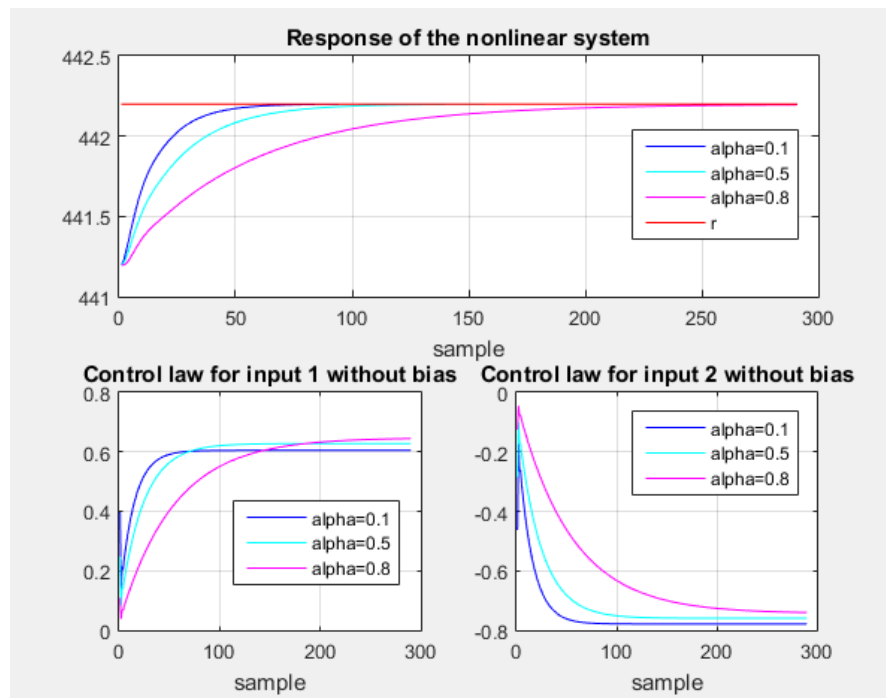


Figure 10. Responses of the system in verifying α .

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

4.4. بررسی پارامتر P یا M :

در این بخش مقدار P و M را که هر دو برابر هستند تغییر می‌دهیم و پاسخ سیستم را به ازای P های 2، 5 و 15 رسم می‌کنیم. شکل زیر پاسخ سیستم به ازای P های مختلف نشان می‌دهد. در این بررسی مقدار آلفا برابر با 0.5 و مقدار گاما برابر با یک قرار داده شده است.

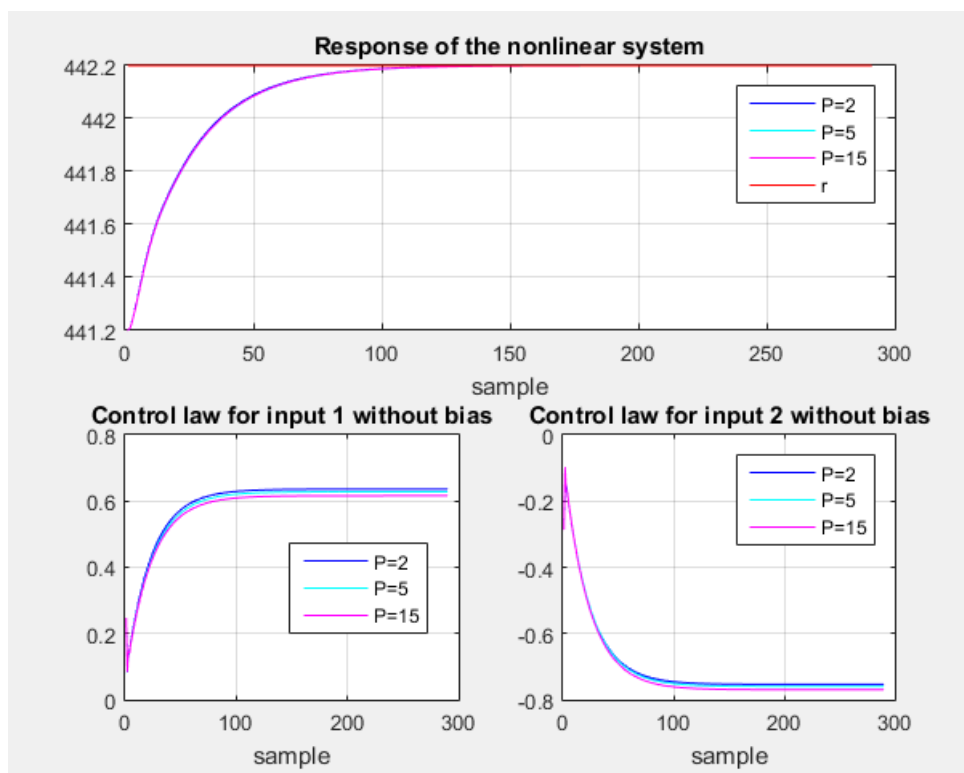


Figure 11. Responses of the system in verifying P or M .

همانطور که در شکل 11 می‌بینیم تغییر پارامتر P یا M تاثیری در پاسخ سیستم نگذاشته است و به ازای هر سه حالت تقریباً پاسخ‌ها روی هم افتاده‌اند که این موضوع با توجه به اینکه همه‌ی پارامترهای کنترل کننده ثابتند، همچنین هیچ اغتشاشی هم نداریم و محدودیتی هم روی سیگنال‌های u نیز وجود ندارد کاملاً طبیعی است.

4.5. 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.

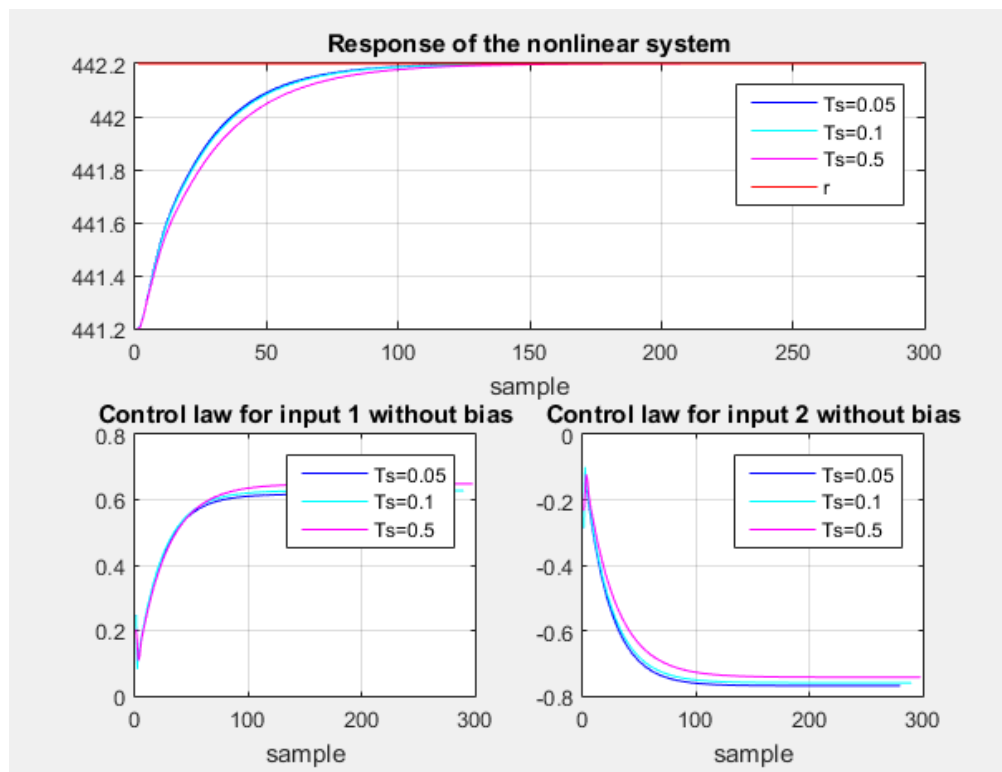


Figure 12. Responses of the system in verifying T_s .

همانطور که در شکل بالا می بینیم وقتی زمان نمونه برداری را کم می کنیم باعث تند شدن پاسخ سیستم می شود. 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.

5. بررسی اثر اغتشاش و نویز:

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

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 می بینیم اثر اغتشاش تنها در لحظه اعمال دیده می شود و در پاسخ حالت ماندگار سیستم تاثیری نمی گذارد و همچنان حتی با وجود اغتشاش خطای حالت ماندگار صفر است.

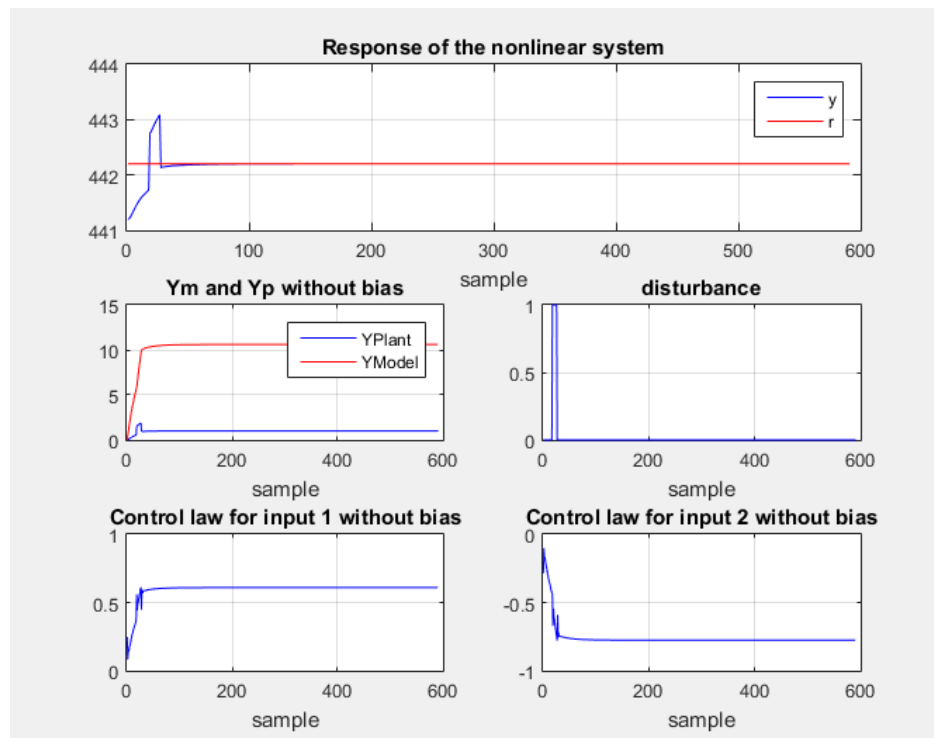


Figure 13. Responses of the system in verifying the effect of disturbance.

5.2. بررسی 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.

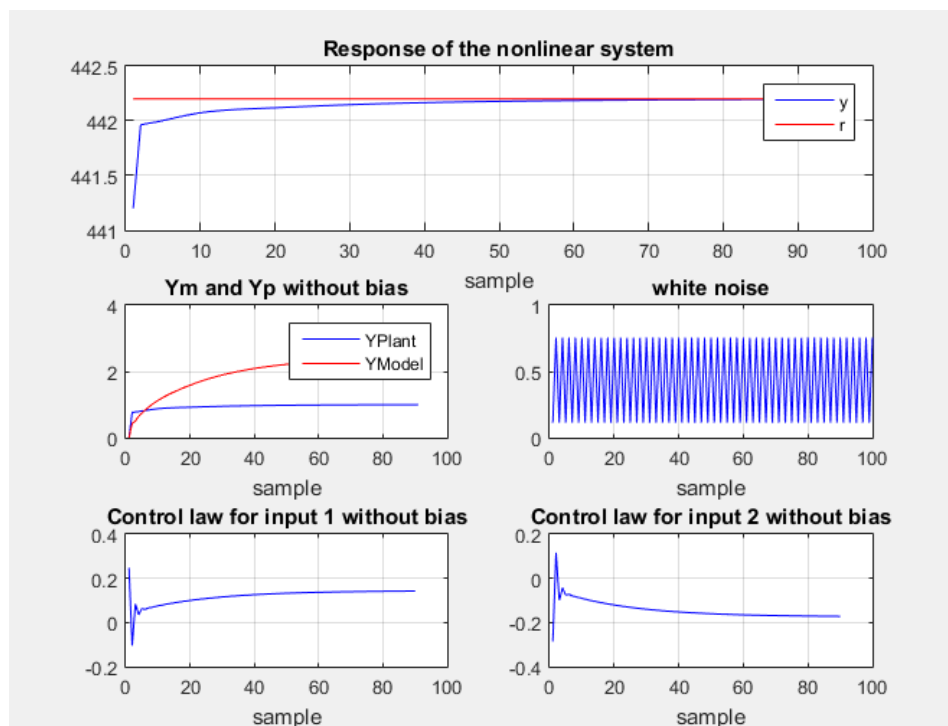


Figure 14. Responses of the system in verifying the effect of noise.

همانطور که در شکل بالا می‌بینیم نویز 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 1% اضافه می‌کنیم. پاسخ سیستم به ازای این عدم قطعیت ها برای این پارامتر در شکل زیر آورده شده است.

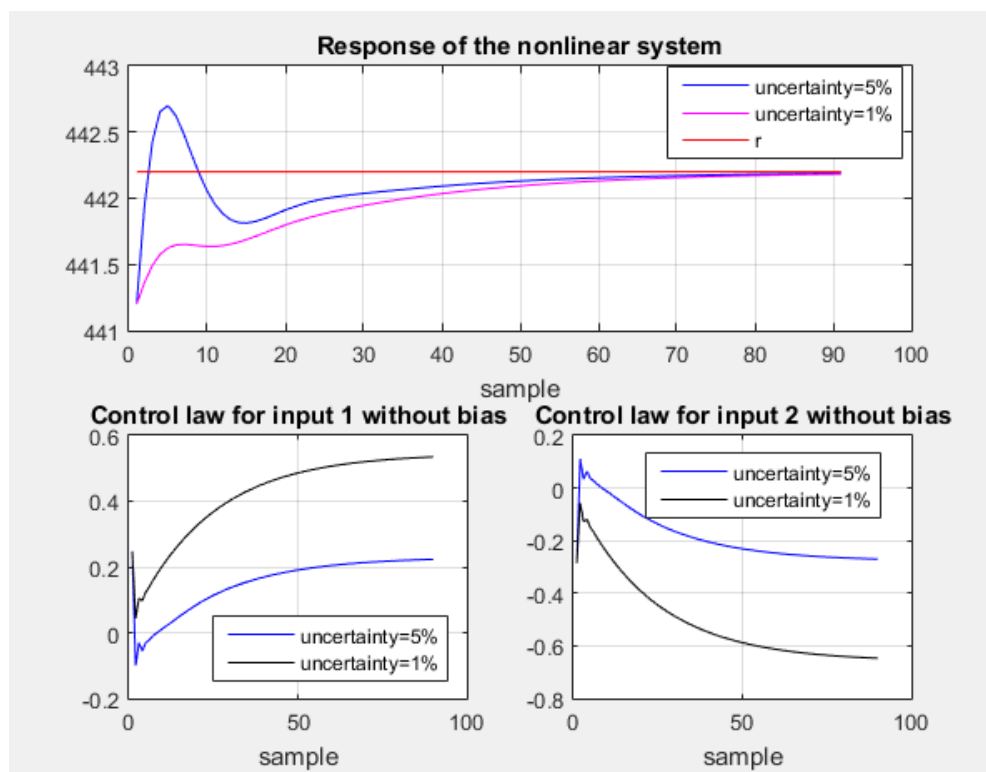


Figure 15. Responses of the system in verifying the effect of uncertainty on V .

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 GPC را نسبت به دو کنترل کننده قبلی را نشان می‌دهد زیرا در دو کنترل کننده قبلی عدم قطعیت بر روی پاسخ حالت ماندگار نیز تاثیر می‌گذاشت.

اینبار عدم قطعیت با درصد های 5% و 1% را به پارامتر C_{A0} اضافه می‌کنیم. شکل زیر پاسخ سیستم در حضور عدم قطعیت های مذکور را نشان می‌دهد.

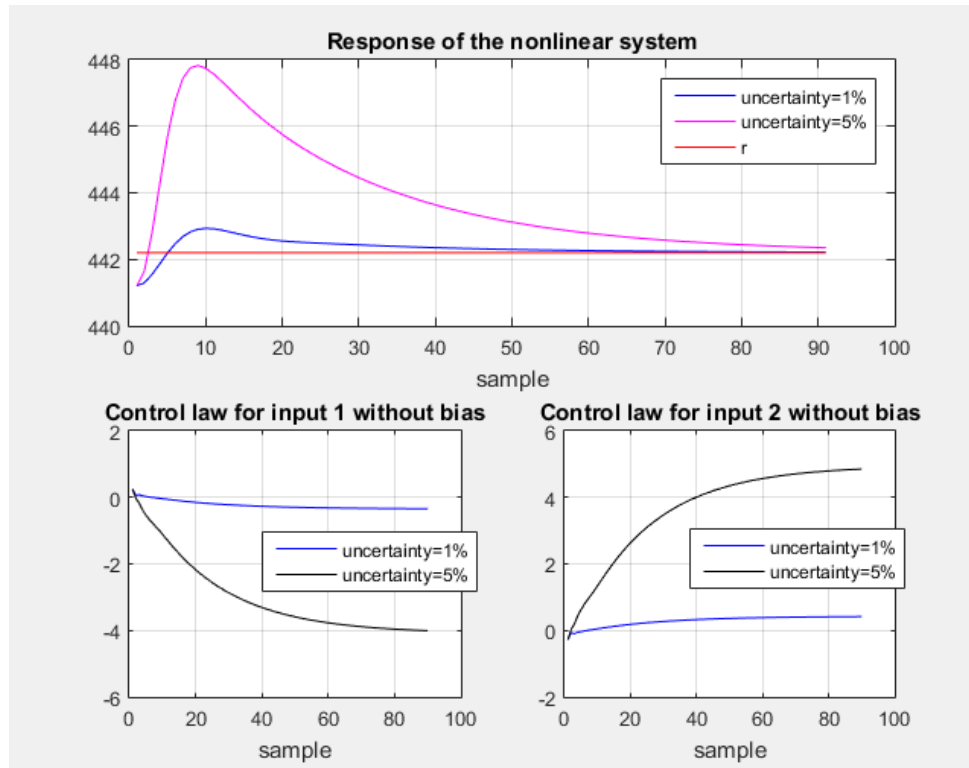


Figure 16. Responses of the system in verifying the effect of uncertainty on CA_0 .

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.

7. کدها و شبیه سازی ها:

کدها و شبیه سازی های 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 برای بررسی پارامترهای مختلف سیستم نوشته شده اند.

% Verifying R

```

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:30;
[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;
%.....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);
G=[G1 G2];
%.....
%A~=-1-2.564z^-1+2.2365z^-2-0.6725z^-3
% According to the discrete transfer function, below parameters have been
% defined
na=3;
nb1=1; nb2=1;
nb=nb1;
d=0;
N1=d+1;
N2=d+P;
a_=[1 -2.564 2.2365 -0.6725];
b1_=num1(2:end);
b2_=num2(2:end);
C=1; % because of using white noise
f=zeros(P+d,na+1);
f(1,1:3)=-1*a_(2:4);
for j=1:P+d-1
    for i=1:na
        f(j+1,i)=f(j,i+1)-f(j,1)*a_(i+1);
    end
end
F=f(N1:N2,1:na);
%.....
E1=zeros(P);
E1(:,1)=1;
for j=1:P-1
    E1(j+1:P,j+1)=f(j,1);
end
B1=zeros(P,P+nb);
for k=1:P
    B1(k,k:k+1)=b1_;
end
m1_=E1*B1;
M1_=zeros(P,nb+d);
for k=1:P
    M1_(k,:)=m1_(k,k+1);
end
%.....

```

```

E2=zeros(P);
E2(:,1)=1;
for j=1:P-1
    E2(j+1:P,j+1)=f(j,1);
end
B2=zeros(P,P+nb);
for k=1:P
    B2(k,k:k+1)=b2_;
end
m2_=E2*B2;
M2_=zeros(P,nb+d);
for k=1:P
    M2_(k,:)=m2_(k,k+1);
end
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(P);
R1=((1.2)^2)*gamma*gain_DC^2*eye(M);
R2=gamma*gain_DC2^2*eye(M);
R=[R1 zeros(M); zeros(M) R2];
alpha=0.5;
Kgpc=(G'*Q*(G+R))/(G'*Q);
x01=0.0882;
x02=441.2;
% .....
dU1_=zeros(nb+d,length(t));
dU2_=zeros(nb+d,length(t));
dU_=[dU1_;dU2_];
d1=zeros(1,length(t));
% y1=0; %linear
y1=441.2;
u_1=[];
u_2=[];
ym=[];
y=0;
Y_d=zeros(P,length(t));
Y_past=zeros(P,length(t));
Y_m=zeros(P,length(t));
D=zeros(P,length(t));
E=zeros(P,length(t));
dU1=zeros(M,length(t));
dU2=zeros(M,length(t));
dU=[dU1;dU2];
U1=zeros(M,length(t));
U2=zeros(M,length(t));
Y_=zeros(na,length(t));
% .....step.....
r =ones(length(t),1);
% .....

for i=1:length(t)-1

```

```

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

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

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

dU(:,i+1)=Kgpc*E(:,i+1);
dU1(:,i+1)=dU(1:M,i+1);
dU2(:,i+1)=dU(M+1:2*M,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:nb+d,i+2) = dU1_(1:nb+d-1,i+1);
dU1_(1,i+2)=dU1(1,i+1);
dU2_(2:nb+d,i+2) = dU2_(1:nb+d-1,i+1);
dU2_(1,i+2)=dU2(1,i+1);
dU_(:,i+2)=[dU1_(:,i+2);dU2_(:,i+2)];
Y_(2:na,i+2)=Y_(1:na-1,i+1);
Y_(1,i+2)=Y_m(1,i+1);

u1=U1(1,i+1);
u2=U2(1,i+1);
sim('Model')
%d(i+2)=yl(end)-Y_m(1,i); %linear
d1(i+2)=y(end)-Y_m(1,i+1);
%y=yl(end); % linear
y=y(end);%+dist(i,1); % nonlinear
%y1=[y1;yl(end)]; % linear
y1=[y1; y+441.2];
ym=[ym; Y_m(1,i)];
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=1/2
gamma =1/2;
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(P);
R1 =((1.2)^2)*gamma*gain_DC^2*eye(M);
R2=gamma*gain_DC2^2*eye(M);
R=[R1 zeros(M); zeros(M) R2];
alpha=0.5;
Kgpc=(G'*Q*(G+R))/(G'*Q);
x01=0.0882;
x02=441.2;
% .....
dU1_=zeros(nb+d,length(t));
dU2_=zeros(nb+d,length(t));
dU_=[dU1_;dU2_];
d1=zeros(1,length(t));
% y1=0; %linear
y1=441.2;
u_1=[];
u_2=[];
ym=[];
y=0;
Y_d=zeros(P,length(t));
Y_past=zeros(P,length(t));
Y_m=zeros(P,length(t));
D=zeros(P,length(t));
E=zeros(P,length(t));
dU1=zeros(M,length(t));
dU2=zeros(M,length(t));
dU=[dU1;dU2];
U1=zeros(M,length(t));
U2=zeros(M,length(t));
Y_=zeros(na,length(t));
% .....step.....
r =ones(length(t),1);
% .....

for i=1:length(t)-1

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

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

```

```

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

dU(:,i+1)=Kgpc*E(:,i+1);
dU1(:,i+1)=dU(1:M,i+1);
dU2(:,i+1)=dU(M+1:2*M,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:nb+d,i+2) = dU1_(1:nb+d-1,i+1);
dU1_(1,i+2)=dU1(1,i+1);
dU2_(2:nb+d,i+2) = dU2_(1:nb+d-1,i+1);
dU2_(1,i+2)=dU2(1,i+1);
dU_(:,i+2)=[dU1_(:,i+2);dU2_(:,i+2)];
Y_(2:na,i+2)=Y_(1:na-1,i+1);
Y_(1,i+2)=Y_m(1,i+1);

u1=U1(1,i+1);
u2=U2(1,i+1);
sim('Model')
%d(i+2)=yl(end)-Y_m(1,i); %linear
d1(i+2)=y(end)-Y_m(1,i+1);
%y=yl(end); % linear
y=y(end);%+dist(i,1); % nonlinear
%y1=[y1;yl(end)]; % linear
y1=[y1; y+441.2];
ym=[ym; Y_m(1,i)];
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=1/600
gamma =1/600;
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(P);
R1=((1.2)^2)*gamma*gain_DC^2*eye(M);
R2=gamma*gain_DC2^2*eye(M);

```

```

R=[R1 zeros(M); zeros(M) R2];
alpha=0.5;
Kgpc=(G'*Q*(G+R)\(G'*Q);
x01=0.0882;
x02=441.2;
% .....
dU1_=zeros(nb+d,length(t));
dU2_=zeros(nb+d,length(t));
dU_=[dU1_;dU2_];
d1=zeros(1,length(t));
% y1=0; % linear
y1=441.2;
u_1=[];
u_2=[];
ym=[];
y=0;
Y_d=zeros(P,length(t));
Y_past=zeros(P,length(t));
Y_m=zeros(P,length(t));
D=zeros(P,length(t));
E=zeros(P,length(t));
dU1=zeros(M,length(t));
dU2=zeros(M,length(t));
dU=[dU1;dU2];
U1=zeros(M,length(t));
U2=zeros(M,length(t));
Y_=zeros(na,length(t));
% .....step.....
r =ones(length(t),1);
% .....

for i=1:length(t)-1

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

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

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

dU(:,i+1)=Kgpc*E(:,i+1);
dU1(:,i+1)=dU(1:M,i+1);
dU2(:,i+1)=dU(M+1:2*M,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:nb+d,i+2) = dU1_(1:nb+d-1,i+1);
dU1_(1,i+2)=dU1(1,i+1);
dU2_(2:nb+d,i+2) = dU2_(1:nb+d-1,i+1);

```

```

dU2_(1,i+2)=dU2(1,i+1);
dU_(:,i+2)=[dU1_(:,i+2);dU2_(:,i+2)];
Y_(2:na,i+2)=Y_(1:na-1,i+1);
Y_(1,i+2)=Y_m(1,i+1);

u1=U1(1,i+1);
u2=U2(1,i+1);
sim('Model')
%d(i+2)=yl(end)-Y_m(1,i); %linear
d1(i+2)=y(end)-Y_m(1,i+1);
%y=yl(end); % linear
y=y(end);%+dist(i,1); % nonlinear
%y1=[y1;yl(end)]; % linear
y1=[y1; y+441.2];
ym=[ym; Y_m(1,i)];
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=1/2','gamma=1/600','r');

subplot(2,2,3);
hold on
plot(u_1,'m');
legend('gamma=1','gamma=1/2','gamma=1/600');
subplot(2,2,4);
hold on
plot(u_2,'m');
legend('gamma=1','gamma=1/2','gamma=1/600');

% Verifying 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:30;
[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;
%.....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);
G=[G1 G2];
%.....
%  $A \sim 1 - 2.564z^{-1} + 2.2365z^{-2} - 0.6725z^{-3}$ 
% According to the discrete transfer function, below parameters have been
% defined
na=3;
nb1=1; nb2=1;
nb=nb1;
d=0;
N1=d+1;
N2=d+P;
a_=[1 -2.564 2.2365 -0.6725];
b1_=num1(2:end);
b2_=num2(2:end);
C=1; % because of using white noise
f=zeros(P+d,na+1);
f(1,1:3)=-1*a_(2:4);
for j=1:P+d-1
    for i=1:na
        f(j+1,i)=f(j,i+1)-f(j,1)*a_(i+1);
    end
end
F=f(N1:N2,1:na);
%.....
E1=zeros(P);
E1(:,1)=1;
for j=1:P-1
    E1(j+1:P,j+1)=f(j,1);
end
B1=zeros(P,P+nb);
for k=1:P
    B1(k,k:k+1)=b1_;

```



```

end
m1_=E1*B1;
M1_=zeros(P,nb+d);
for k=1:P
    M1_(k,:)=m1_(k,k+1);
end
%.....
E2=zeros(P);
E2(:,1)=1;
for j=1:P-1
    E2(j+1:P,j+1)=f(j,1);
end
B2=zeros(P,P+nb);
for k=1:P
    B2(k,k:k+1)=b2_;
end
m2_=E2*B2;
M2_=zeros(P,nb+d);
for k=1:P
    M2_(k,:)=m2_(k,k+1);
end
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(P);
R1 =((1.2)^2)*gamma*gain_DC^2*eye(M);
R2=gamma*gain_DC2^2*eye(M);
R=[R1 zeros(M); zeros(M) R2];
alpha=0.5;
Kgpc=(G'*Q*(G+R))/(G'*Q);
x01=0.0882;
x02=441.2;
%.....
dU1_=zeros(nb+d,length(t));
dU2_=zeros(nb+d,length(t));
dU_=[dU1_;dU2_];
d1=zeros(1,length(t));
%y1=0; %linear
y1=441.2;
u_1=[];
u_2=[];
ym=[];
y=0;
Y_d=zeros(P,length(t));
Y_past=zeros(P,length(t));
Y_m=zeros(P,length(t));
D=zeros(P,length(t));
E=zeros(P,length(t));
dU1=zeros(M,length(t));
dU2=zeros(M,length(t));
dU=[dU1;dU2];
U1=zeros(M,length(t));
U2=zeros(M,length(t));

```

```

Y_=zeros(na,length(t));
%.....step.....
r=ones(length(t),1);
%.....

for i=1:length(t)-1

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

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

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

dU(:,i+1)=Kgpc*E(:,i+1);
dU1(:,i+1)=dU(1:M,i+1);
dU2(:,i+1)=dU(M+1:2*M,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:nb+d,i+2)=dU1_(1:nb+d-1,i+1);
dU1_(1,i+2)=dU1(1,i+1);
dU2_(2:nb+d,i+2)=dU2_(1:nb+d-1,i+1);
dU2_(1,i+2)=dU2(1,i+1);
dU_(:,i+2)=[dU1_(:,i+2);dU2_(:,i+2)];
Y_(2:na,i+2)=Y_(1:na-1,i+1);
Y_(1,i+2)=Y_m(1,i+1);

u1=U1(1,i+1);
u2=U2(1,i+1);
sim('Model')
%d(i+2)=yl(end)-Y_m(1,i); % linear
d1(i+2)=y(end)-Y_m(1,i+1);
%y=yl(end); % linear
y=y(end);%+dist(i,1); % nonlinear
%y1=[y1;yl(end)]; % linear
y1=[y1; y+441.2];
ym=[ym; Y_m(1,i)];
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=2I
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 = 2*eye(P);
R1 = ((1.2)^2)*gamma*gain_DC^2*eye(M);
R2=gamma*gain_DC2^2*eye(M);
R=[R1 zeros(M); zeros(M) R2];
alpha=0.5;
Kgpc=(G'*Q*G+R)\(G'*Q);
x01=0.0882;
x02=441.2;
% .....
dU1_=zeros(nb+d,length(t));
dU2_=zeros(nb+d,length(t));
dU_=[dU1_;dU2_];
d1=zeros(1,length(t));
%y1=0; %linear
y1=441.2;
u_1=[];
u_2=[];
ym=[];
y=0;
Y_d=zeros(P,length(t));
Y_past=zeros(P,length(t));
Y_m=zeros(P,length(t));
D=zeros(P,length(t));
E=zeros(P,length(t));
dU1=zeros(M,length(t));
dU2=zeros(M,length(t));
dU=[dU1;dU2];
U1=zeros(M,length(t));
U2=zeros(M,length(t));
Y_=zeros(na,length(t));
% .....step.....
r =ones(length(t),1);
% .....

for i=1:length(t)-1

for j=1:P

```

```

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

```

```

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

```

```

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

```

```

dU(:,i+1)=Kgpc*E(:,i+1);
dU1(:,i+1)=dU(1:M,i+1);
dU2(:,i+1)=dU(M+1:2*M,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:nb+d,i+2) = dU1_(1:nb+d-1,i+1);
dU1_(1,i+2)=dU1(1,i+1);
dU2_(2:nb+d,i+2) = dU2_(1:nb+d-1,i+1);
dU2_(1,i+2)=dU2(1,i+1);
dU_(:,i+2)=[dU1_(:,i+2);dU2_(:,i+2)];
Y_(2:na,i+2)=Y_(1:na-1,i+1);
Y_(1,i+2)=Y_m(1,i+1);

```

```

u1=U1(1,i+1);
u2=U2(1,i+1);
sim('Model')
%d(i+2)=yl(end)-Y_m(1,i); %linear
d1(i+2)=y(end)-Y_m(1,i+1);
%y=yl(end); % linear
y=y(end);%+dist(i,1); % nonlinear
%y1=[y1;yl(end)]; % linear
y1=[y1; y+441.2];
ym=[ym; Y_m(1,i)];
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=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(P);
R1 =((1.2)^2)*gamma*gain_DC^2*eye(M);
R2=gamma*gain_DC2^2*eye(M);
R=[R1 zeros(M); zeros(M) R2];
alpha=0.5;
Kgpc=(G'*Q*G+R)\(G'*Q);
x01=0.0882;
x02=441.2;
% .....
dU1_=zeros(nb+d,length(t));
dU2_=zeros(nb+d,length(t));
dU_=[dU1_;dU2_];
d1=zeros(1,length(t));
%y1=0; %linear
y1=441.2;
u_1=[];
u_2=[];
ym=[];
y=0;
Y_d=zeros(P,length(t));
Y_past=zeros(P,length(t));
Y_m=zeros(P,length(t));
D=zeros(P,length(t));
E=zeros(P,length(t));
dU1=zeros(M,length(t));
dU2=zeros(M,length(t));
dU=[dU1;dU2];
U1=zeros(M,length(t));
U2=zeros(M,length(t));
Y_=zeros(na,length(t));
% .....step.....
r =ones(length(t),1);
% .....

for i=1:length(t)-1

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

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

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

dU(:,i+1)=Kgpc*E(:,i+1);
dU1(:,i+1)=dU(1:M,i+1);
dU2(:,i+1)=dU(M+1:2*M,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:nb+d,i+2) = dU1_(1:nb+d-1,i+1);
dU1_(1,i+2)=dU1(1,i+1);
dU2_(2:nb+d,i+2) = dU2_(1:nb+d-1,i+1);
dU2_(1,i+2)=dU2(1,i+1);
dU_(:,i+2)=[dU1_(:,i+2);dU2_(:,i+2)];
Y_(2:na,i+2)=Y_(1:na-1,i+1);
Y_(1,i+2)=Y_m(1,i+1);
```

```
u1=U1(1,i+1);
u2=U2(1,i+1);
sim('Model')
%d(i+2)=yl(end)-Y_m(1,i); %linear
d1(i+2)=y(end)-Y_m(1,i+1);
%y=yl(end); % linear
y=y(end);%+dist(i,1); % nonlinear
%y1=[y1;yl(end)]; % linear
y1=[y1; y+441.2];
ym=[ym; Y_m(1,i)];
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=2I','Q=60I','r');
subplot(2,2,3);
hold on
plot(u_1,'m');
legend('Q=I','Q=2I','Q=60I');
subplot(2,2,4);
hold on
plot(u_2,'m');
legend('Q=I','Q=2I','Q=60I');
```

```
% Verifying 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:30;
[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;
%.....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);
G=[G1 G2];
%.....
%  $A \sim 1 - 2.564z^{-1} + 2.2365z^{-2} - 0.6725z^{-3}$ 
% According to the discrete transfer function, below parameters have been
% defined
na=3;
nb1=1; nb2=1;
nb=nb1;
d=0;
N1=d+1;
N2=d+P;
a_=[1 -2.564 2.2365 -0.6725];
b1_=num1(2:end);
b2_=num2(2:end);
C=1; % because of using white noise
f=zeros(P+d,na+1);
f(1,1:3)=-1*a_(2:4);
for j=1:P+d-1
    for i=1:na
        f(j+1,i)=f(j,i+1)-f(j,1)*a_(i+1);
    end
end
F=f(N1:N2,1:na);
%.....
E1=zeros(P);
E1(:,1)=1;
for j=1:P-1

```

```

    E1(j+1:P,j+1)=f(j,1);
end
B1=zeros(P,P+nb);
for k=1:P
    B1(k,k:k+1)=b1_;
end
m1_=E1*B1;
M1_=zeros(P,nb+d);
for k=1:P
    M1_(k,:)=m1_(k,k+1);
end
% .....
E2=zeros(P);
E2(:,1)=1;
for j=1:P-1
    E2(j+1:P,j+1)=f(j,1);
end
B2=zeros(P,P+nb);
for k=1:P
    B2(k,k:k+1)=b2_;
end
m2_=E2*B2;
M2_=zeros(P,nb+d);
for k=1:P
    M2_(k,:)=m2_(k,k+1);
end
M_=[M1_ M2_];
% .....
% alpha=0.1
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(P);
R1=((1.2)^2)*gamma*gain_DC^2*eye(M);
R2=gamma*gain_DC2^2*eye(M);
R=[R1 zeros(M); zeros(M) R2];
alpha=0.1;
Kgpc=(G'*Q*(G+R))/(G'*Q);
x01=0.0882;
x02=441.2;
% .....
dU1_=zeros(nb+d,length(t));
dU2_=zeros(nb+d,length(t));
dU_=[dU1_;dU2_];
d1=zeros(1,length(t));
% y1=0; %linear
y1=441.2;
u_1=[];
u_2=[];
ym=[];
y=0;
Y_d=zeros(P,length(t));
Y_past=zeros(P,length(t));
Y_m=zeros(P,length(t));
D=zeros(P,length(t));
E=zeros(P,length(t));

```



```

dU1=zeros(M,length(t));
dU2=zeros(M,length(t));
dU=[dU1;dU2];
U1=zeros(M,length(t));
U2=zeros(M,length(t));
Y_=zeros(na,length(t));
%.....step.....
r =ones(length(t),1);
%.....

for i=1:length(t)-1

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

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

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

dU(:,i+1)=Kgpc*E(:,i+1);
dU1(:,i+1)=dU(1:M,i+1);
dU2(:,i+1)=dU(M+1:2*M,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:nb+d,i+2) = dU1_(1:nb+d-1,i+1);
dU1_(1,i+2)=dU1(1,i+1);
dU2_(2:nb+d,i+2) = dU2_(1:nb+d-1,i+1);
dU2_(1,i+2)=dU2(1,i+1);
dU_(:,i+2)=[dU1_(:,i+2);dU2_(:,i+2)];
Y_(2:na,i+2)=Y_(1:na-1,i+1);
Y_(1,i+2)=Y_m(1,i+1);

u1=U1(1,i+1);
u2=U2(1,i+1);
sim('Model')
%d(i+2)=yl(end)-Y_m(1,i); %linear
d1(i+2)=y(end)-Y_m(1,i+1);
%y=yl(end); % linear
y=y(end);%+dist(i,1); % nonlinear
%y1=[y1;yl(end)]; % linear
y1=[y1; y+441.2];
ym=[ym; Y_m(1,i)];
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=0.5
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(P);
R1 =((1.2)^2)*gamma*gain_DC^2*eye(M);
R2=gamma*gain_DC2^2*eye(M);
R=[R1 zeros(M); zeros(M) R2];
alpha=0.5;
Kgpc=(G'*Q*G+R)\(G'*Q);
x01=0.0882;
x02=441.2;
% .....
dU1_=zeros(nb+d,length(t));
dU2_=zeros(nb+d,length(t));
dU_=[dU1_;dU2_];
d1=zeros(1,length(t));
%y1=0; %linear
y1=441.2;
u_1=[];
u_2=[];
ym=[];
y=0;
Y_d=zeros(P,length(t));
Y_past=zeros(P,length(t));
Y_m=zeros(P,length(t));
D=zeros(P,length(t));
E=zeros(P,length(t));
dU1=zeros(M,length(t));
dU2=zeros(M,length(t));
dU=[dU1;dU2];
U1=zeros(M,length(t));
U2=zeros(M,length(t));
Y_=zeros(na,length(t));
% .....step.....
r =ones(length(t),1);
% .....
```

```

for i=1:length(t)-1

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

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

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

dU(:,i+1)=Kgpc*E(:,i+1);
dU1(:,i+1)=dU(1:M,i+1);
dU2(:,i+1)=dU(M+1:2*M,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:nb+d,i+2) = dU1_(1:nb+d-1,i+1);
dU1_(1,i+2)=dU1(1,i+1);
dU2_(2:nb+d,i+2) = dU2_(1:nb+d-1,i+1);
dU2_(1,i+2)=dU2(1,i+1);
dU_(:,i+2)=[dU1_(:,i+2);dU2_(:,i+2)];
Y_(2:na,i+2)=Y_(1:na-1,i+1);
Y_(1,i+2)=Y_m(1,i+1);

u1=U1(1,i+1);
u2=U2(1,i+1);
sim('Model')
%d(i+2)=yl(end)-Y_m(1,i); %linear
d1(i+2)=y(end)-Y_m(1,i+1);
%y=yl(end); % linear
y=y(end);%+dist(i,1); % nonlinear
%y1=[y1;yl(end)]; % linear
y1=[y1; y+441.2];
ym=[ym; Y_m(1,i)];
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;
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(P);
R1=((1.2)^2)*gamma*gain_DC^2*eye(M);
R2=gamma*gain_DC2^2*eye(M);
R=[R1 zeros(M); zeros(M) R2];
alpha=0.8;
Kgpc=(G'*Q*(G+R))/(G'*Q);
x01=0.0882;
x02=441.2;
% .....
dU1_=zeros(nb+d,length(t));
dU2_=zeros(nb+d,length(t));
dU_=[dU1_;dU2_];
d1=zeros(1,length(t));
% y1=0; % linear
y1=441.2;
u_1=[];
u_2=[];
ym=[];
y=0;
Y_d=zeros(P,length(t));
Y_past=zeros(P,length(t));
Y_m=zeros(P,length(t));
D=zeros(P,length(t));
E=zeros(P,length(t));
dU1=zeros(M,length(t));
dU2=zeros(M,length(t));
dU=[dU1;dU2];
U1=zeros(M,length(t));
U2=zeros(M,length(t));
Y_=zeros(na,length(t));
% ..... step .....
r =ones(length(t),1);
% .....

for i=1:length(t)-1

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

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

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

dU(:,i+1)=Kgpc*E(:,i+1);
dU1(:,i+1)=dU(1:M,i+1);

```

```

dU2(:,i+1)=dU(M+1:2*M,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:nb+d,i+2) = dU1_(1:nb+d-1,i+1);
dU1_(1,i+2)=dU1(1,i+1);
dU2_(2:nb+d,i+2) = dU2_(1:nb+d-1,i+1);
dU2_(1,i+2)=dU2(1,i+1);
dU_(:,i+2)=[dU1_(:,i+2);dU2_(:,i+2)];
Y_(2:na,i+2)=Y_(1:na-1,i+1);
Y_(1,i+2)=Y_m(1,i+1);

u1=U1(1,i+1);
u2=U2(1,i+1);
sim('Model')
%d(i+2)=yl(end)-Y_m(1,i); %linear
d1(i+2)=y(end)-Y_m(1,i+1);
%y=yl(end); % linear
y=y(end);%+dist(i,1); % nonlinear
%y1=[y1;yl(end)]; % linear
y1=[y1; y+441.2];
ym=[ym; Y_m(1,i)];
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=0.1','alpha=0.5','alpha=0.8','r');
subplot(2,2,3);
hold on
plot(u_1,'m');
legend('alpha=0.1','alpha=0.5','alpha=0.8');
subplot(2,2,4);
hold on
plot(u_2,'m');
legend('alpha=0.1','alpha=0.5','alpha=0.8');

% Verifying P or M
% P=2
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:30;
[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=2;
% P=max(P1,P2);
N=max(N1,N2);
M=P;
% .....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);
G=[G1 G2];
% .....
% A~=-1-2.564z^-1+2.2365z^-2-0.6725z^-3
% According to the discrete transfer function, below parameters have been
% defined
na=3;
nb1=1; nb2=1;
nb=nb1;
d=0;
N1=d+1;
N2=d+P;
a_=[1 -2.564 2.2365 -0.6725];
b1_=num1(2:end);
b2_=num2(2:end);
C=1; % because of using white noise
f=zeros(P+d,na+1);
f(1,1:3)=-1*a_(2:4);
for j=1:P+d-1
    for i=1:na
        f(j+1,i)=f(j,i+1)-f(j,1)*a_(i+1);
    end
end

```

```

end
F=f(N1:N2,1:na);
%.....
E1=zeros(P);
E1(:,1)=1;
for j=1:P-1
    E1(j+1:P,j+1)=f(j,1);
end
B1=zeros(P,P+nb);
for k=1:P
    B1(k,k:k+1)=b1_;
end
m1_=E1*B1;
M1_=zeros(P,nb+d);
for k=1:P
    M1_(k,:)=m1_(k,k+1);
end
%.....
E2=zeros(P);
E2(:,1)=1;
for j=1:P-1
    E2(j+1:P,j+1)=f(j,1);
end
B2=zeros(P,P+nb);
for k=1:P
    B2(k,k:k+1)=b2_;
end
m2_=E2*B2;
M2_=zeros(P,nb+d);
for k=1:P
    M2_(k,:)=m2_(k,k+1);
end
M_=[M1_ M2_];
%.....
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(P);
R1 =((1.2)^2)*gamma*gain_DC^2*eye(M);
R2=gamma*gain_DC2^2*eye(M);
R=[R1 zeros(M); zeros(M) R2];
alpha=0.5;
Kgpc=(G'*Q*(G+R))/(G'*Q);
x01=0.0882;
x02=441.2;
%.....
dU1_=zeros(nb+d,length(t));
dU2_=zeros(nb+d,length(t));
dU_=[dU1_;dU2_];
d1=zeros(1,length(t));
%y1=0; %linear
y1=441.2;
u_1=[];
u_2=[];
ym=[];
y=0;

```

```

Y_d=zeros(P,length(t));
Y_past=zeros(P,length(t));
Y_m=zeros(P,length(t));
D=zeros(P,length(t));
E=zeros(P,length(t));
dU1=zeros(M,length(t));
dU2=zeros(M,length(t));
dU=[dU1;dU2];
U1=zeros(M,length(t));
U2=zeros(M,length(t));
Y_=zeros(na,length(t));
% .....step.....
r =ones(length(t),1);
% .....

for i=1:length(t)-1

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

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

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

dU(:,i+1)=Kgpc*E(:,i+1);
dU1(:,i+1)=dU(1:M,i+1);
dU2(:,i+1)=dU(M+1:2*M,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:nb+d,i+2) = dU1_(1:nb+d-1,i+1);
dU1_(1,i+2)=dU1(1,i+1);
dU2_(2:nb+d,i+2) = dU2_(1:nb+d-1,i+1);
dU2_(1,i+2)=dU2(1,i+1);
dU_(:,i+2)=[dU1_(:,i+2);dU2_(:,i+2)];
Y_(2:na,i+2)=Y_(1:na-1,i+1);
Y_(1,i+2)=Y_m(1,i+1);

u1=U1(1,i+1);
u2=U2(1,i+1);
sim('Model')
% d(i+2)=yl(end)-Y_m(1,i); % linear
d1(i+2)=y(end)-Y_m(1,i+1);
% y=yl(end); % linear
y=y(end); % dist(i,1); % nonlinear
% y1=[y1;yl(end)]; % linear
y1=[y1; y+441.2];
ym=[ym; Y_m(1,i)];
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');
% .....
% P=5
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:30;
[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;
% .....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);
G=[G1 G2];
%.....
%A~=-1-2.564z^-1+2.2365z^-2-0.6725z^-3
% According to the discrete transfer function, below parameters have been
% defined
na=3;
nb1=1; nb2=1;
nb=nb1;
d=0;
N1=d+1;
N2=d+P;
a_=[1 -2.564 2.2365 -0.6725];
b1_=num1(2:end);
b2_=num2(2:end);
C=1; % because of using white noise
f=zeros(P+d,na+1);
f(1,1:3)=-1*a_(2:4);
for j=1:P+d-1
    for i=1:na
        f(j+1,i)=f(j,i+1)-f(j,1)*a_(i+1);
    end
end
F=f(N1:N2,1:na);
%.....
E1=zeros(P);
E1(:,1)=1;
for j=1:P-1
    E1(j+1:P,j+1)=f(j,1);
end
B1=zeros(P,P+nb);
for k=1:P
    B1(k,k:k+1)=b1_;
end
m1_=E1*B1;
M1_=zeros(P,nb+d);
for k=1:P
    M1_(k,:)=m1_(k,k+1);
end
%.....
E2=zeros(P);
E2(:,1)=1;
for j=1:P-1
    E2(j+1:P,j+1)=f(j,1);
end
B2=zeros(P,P+nb);
for k=1:P
    B2(k,k:k+1)=b2_;
end

```

```

m2_=E2*B2;
M2_=zeros(P,nb+d);
for k=1:P
    M2_(k,:)=m2_(k,k+1);
end
M_=[M1_ M2_];
% .....
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(P);
R1 =((1.2)^2)*gamma*gain_DC^2*eye(M);
R2=gamma*gain_DC2^2*eye(M);
R=[R1 zeros(M); zeros(M) R2];
alpha=0.5;
Kgpc=(G'*Q*(G+R))/(G'*Q);
x01=0.0882;
x02=441.2;
% .....
dU1_=zeros(nb+d,length(t));
dU2_=zeros(nb+d,length(t));
dU_=[dU1_;dU2_];
d1=zeros(1,length(t));
% y1=0; % linear
y1=441.2;
u_1=[];
u_2=[];
ym=[];
y=0;
Y_d=zeros(P,length(t));
Y_past=zeros(P,length(t));
Y_m=zeros(P,length(t));
D=zeros(P,length(t));
E=zeros(P,length(t));
dU1=zeros(M,length(t));
dU2=zeros(M,length(t));
dU=[dU1;dU2];
U1=zeros(M,length(t));
U2=zeros(M,length(t));
Y_=zeros(na,length(t));
% .....step.....
r =ones(length(t),1);
% .....

for i=1:length(t)-1

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

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

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

```

```

dU(:,i+1)=Kgpc*E(:,i+1);
dU1(:,i+1)=dU(1:M,i+1);
dU2(:,i+1)=dU(M+1:2*M,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:nb+d,i+2) = dU1_(1:nb+d-1,i+1);
dU1_(1,i+2)=dU1(1,i+1);
dU2_(2:nb+d,i+2) = dU2_(1:nb+d-1,i+1);
dU2_(1,i+2)=dU2(1,i+1);
dU_(:,i+2)=[dU1_(:,i+2);dU2_(:,i+2)];
Y_(2:na,i+2)=Y_(1:na-1,i+1);
Y_(1,i+2)=Y_m(1,i+1);

u1=U1(1,i+1);
u2=U2(1,i+1);
sim('Model')
%d(i+2)=yl(end)-Y_m(1,i); %linear
d1(i+2)=y(end)-Y_m(1,i+1);
%y=yl(end); % linear
y=y(end);%+dist(i,1); % nonlinear
%y1=[y1;yl(end)]; % linear
y1=[y1; y+441.2];
ym=[ym; Y_m(1,i)];
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');
% .....
% P=15
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:30;
[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=15
N=max(N1,N2);
M=P;
%.....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);
G=[G1 G2];
%.....
%A~=-1-2.564z^-1+2.2365z^-2-0.6725z^-3
% According to the discrete transfer function, below parameters have been
% defined
na=3;
nb1=1; nb2=1;
nb=nb1;
d=0;
N1=d+1;
N2=d+P;
a_=[1 -2.564 2.2365 -0.6725];
b1_=num1(2:end);
b2_=num2(2:end);
C=1; % because of using white noise
f=zeros(P+d,na+1);
f(1,1:3)=-1*a_(2:4);
for j=1:P+d-1
    for i=1:na
        f(j+1,i)=f(j,i+1)-f(j,1)*a_(i+1);
    end
end
F=f(N1:N2,1:na);
%.....
E1=zeros(P);
E1(:,1)=1;

```

```

for j=1:P-1
    E1(j+1:P,j+1)=f(j,1);
end
B1=zeros(P,P+nb);
for k=1:P
    B1(k,k:k+1)=b1_;
end
m1_=E1*B1;
M1_=zeros(P,nb+d);
for k=1:P
    M1_(k,:)=m1_(k,k+1);
end
% .....
E2=zeros(P);
E2(:,1)=1;
for j=1:P-1
    E2(j+1:P,j+1)=f(j,1);
end
B2=zeros(P,P+nb);
for k=1:P
    B2(k,k:k+1)=b2_;
end
m2_=E2*B2;
M2_=zeros(P,nb+d);
for k=1:P
    M2_(k,:)=m2_(k,k+1);
end
M_=[M1_ M2_];
% .....
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(P);
R1=((1.2)^2)*gamma*gain_DC^2*eye(M);
R2=gamma*gain_DC2^2*eye(M);
R=[R1 zeros(M); zeros(M) R2];
alpha=0.5;
Kgpc=(G'*Q*(G+R))/(G'*Q);
x01=0.0882;
x02=441.2;
% .....
dU1_=zeros(nb+d,length(t));
dU2_=zeros(nb+d,length(t));
dU_=[dU1_;dU2_];
d1=zeros(1,length(t));
% y1=0; %linear
y1=441.2;
u_1=[];
u_2=[];
ym=[];
y=0;
Y_d=zeros(P,length(t));
Y_past=zeros(P,length(t));
Y_m=zeros(P,length(t));
D=zeros(P,length(t));
E=zeros(P,length(t));

```

```

dU1=zeros(M,length(t));
dU2=zeros(M,length(t));
dU=[dU1;dU2];
U1=zeros(M,length(t));
U2=zeros(M,length(t));
Y_=zeros(na,length(t));
%.....step.....
r =ones(length(t),1);
%.....

for i=1:length(t)-1

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

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

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

dU(:,i+1)=Kgpc*E(:,i+1);
dU1(:,i+1)=dU(1:M,i+1);
dU2(:,i+1)=dU(M+1:2*M,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:nb+d,i+2) = dU1_(1:nb+d-1,i+1);
dU1_(1,i+2)=dU1(1,i+1);
dU2_(2:nb+d,i+2) = dU2_(1:nb+d-1,i+1);
dU2_(1,i+2)=dU2(1,i+1);
dU_(:,i+2)=[dU1_(:,i+2);dU2_(:,i+2)];
Y_(2:na,i+2)=Y_(1:na-1,i+1);
Y_(1,i+2)=Y_m(1,i+1);

u1=U1(1,i+1);
u2=U2(1,i+1);
sim('Model')
%d(i+2)=yl(end)-Y_m(1,i); %linear
d1(i+2)=y(end)-Y_m(1,i+1);
%y=yl(end); % linear
y=y(end);%+dist(i,1); % nonlinear
%y1=[y1;yl(end)]; % linear
y1=[y1; y+441.2];
ym=[ym; Y_m(1,i)];
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('P=2','P=5','P=15','r');
subplot(2,2,3);
hold on
plot(u_1,'m');
legend('P=2','P=5','P=15');
subplot(2,2,4);
hold on
plot(u_2,'m');
legend('P=2','P=5','P=15');

```

```

% Verifying 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:15;
[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;
%.....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);

```



```

G=[G1 G2];
% .....
%  $A \sim 1 - 2.564z^{-1} + 2.2365z^{-2} - 0.6725z^{-3}$ 
% According to the discrete transfer function, below parameters have been
% defined
na=3;
nb1=1; nb2=1;
nb=nb1;
d=0;
N1=d+1;
N2=d+P;
a_=[1 -2.564 2.2365 -0.6725];
b1_=num1(2:end);
b2_=num2(2:end);
C=1; % because of using white noise
f=zeros(P+d,na+1);
f(1,1:3)=-1*a_(2:4);
for j=1:P+d-1
    for i=1:na
        f(j+1,i)=f(j,i+1)-f(j,1)*a_(i+1);
    end
end
F=f(N1:N2,1:na);
% .....
E1=zeros(P);
E1(:,1)=1;
for j=1:P-1
    E1(j+1:P,j+1)=f(j,1);
end
B1=zeros(P,P+nb);
for k=1:P
    B1(k,k:k+1)=b1_;
end
m1_=E1*B1;
M1_=zeros(P,nb+d);
for k=1:P
    M1_(k,:)=m1_(k,k+1);
end
% .....
E2=zeros(P);
E2(:,1)=1;
for j=1:P-1
    E2(j+1:P,j+1)=f(j,1);
end
B2=zeros(P,P+nb);
for k=1:P
    B2(k,k:k+1)=b2_;
end
m2_=E2*B2;
M2_=zeros(P,nb+d);
for k=1:P
    M2_(k,:)=m2_(k,k+1);
end
M_=[M1_ M2_];
% .....
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(P);
R1 =((1.2)^2)*gamma*gain_DC^2*eye(M);
R2=gamma*gain_DC2^2*eye(M);
R=[R1 zeros(M); zeros(M) R2];
alpha=0.5;
Kgpc=(G'*Q*(G+R))/(G'*Q);
x01=0.0882;
x02=441.2;
%.....
dU1_=zeros(nb+d,length(t));
dU2_=zeros(nb+d,length(t));
dU_=[dU1_;dU2_];
d1=zeros(1,length(t));
%y1=0; %linear
y1=441.2;
u_1=[];
u_2=[];
ym=[];
y=0;
Y_d=zeros(P,length(t));
Y_past=zeros(P,length(t));
Y_m=zeros(P,length(t));
D=zeros(P,length(t));
E=zeros(P,length(t));
dU1=zeros(M,length(t));
dU2=zeros(M,length(t));
dU=[dU1;dU2];
U1=zeros(M,length(t));
U2=zeros(M,length(t));
Y_=zeros(na,length(t));
%.....step.....
r =ones(length(t),1);
%.....

for i=1:length(t)-1

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

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

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

dU(:,i+1)=Kgpc*E(:,i+1);
dU1(:,i+1)=dU(1:M,i+1);
dU2(:,i+1)=dU(M+1:2*M,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:nb+d,i+2) = dU1_(1:nb+d-1,i+1);
dU1_(1,i+2)=dU1(1,i+1);
dU2_(2:nb+d,i+2) = dU2_(1:nb+d-1,i+1);
dU2_(1,i+2)=dU2(1,i+1);
dU_(:,i+2)=[dU1_(:,i+2);dU2_(:,i+2)];
Y_(2:na,i+2)=Y_(1:na-1,i+1);
Y_(1,i+2)=Y_m(1,i+1);

```

```

u1=U1(1,i+1);
u2=U2(1,i+1);
sim('Model')
%d(i+2)=yl(end)-Y_m(1,i); %linear
d1(i+2)=y(end)-Y_m(1,i+1);
%y=yl(end); % linear
y=y(end);%+dist(i,1); % nonlinear
%y1=[y1;yl(end)]; % linear
y1=[y1; y+441.2];
ym=[ym; Y_m(1,i)];
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:30;
[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;
%.....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);
G=[G1 G2];
%.....
% A~=-1-2.564z^-1+2.2365z^-2-0.6725z^-3
% According to the discrete transfer function, below parameters have been
% defined
na=3;
nb1=1; nb2=1;
nb=nb1;
d=0;
N1=d+1;
N2=d+P;
a_=[1 -2.564 2.2365 -0.6725];
b1_=num1(2:end);
b2_=num2(2:end);
C=1; % because of using white noise
f=zeros(P+d,na+1);
f(1,1:3)=-1*a_(2:4);
for j=1:P+d-1
    for i=1:na
        f(j+1,i)=f(j,i+1)-f(j,1)*a_(i+1);
    end
end
F=f(N1:N2,1:na);
%.....
E1=zeros(P);
E1(:,1)=1;
for j=1:P-1
    E1(j+1:P,j+1)=f(j,1);

```

```

end
B1=zeros(P,P+nb);
for k=1:P
    B1(k,k:k+1)=b1_;
end
m1_=E1*B1;
M1_=zeros(P,nb+d);
for k=1:P
    M1_(k,:)=m1_(k,k+1);
end
%.....
E2=zeros(P);
E2(:,1)=1;
for j=1:P-1
    E2(j+1:P,j+1)=f(j,1);
end
B2=zeros(P,P+nb);
for k=1:P
    B2(k,k:k+1)=b2_;
end
m2_=E2*B2;
M2_=zeros(P,nb+d);
for k=1:P
    M2_(k,:)=m2_(k,k+1);
end
M_=[M1_ M2_];
%.....
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(P);
R1 =((1.2)^2)*gamma*gain_DC^2*eye(M);
R2=gamma*gain_DC2^2*eye(M);
R=[R1 zeros(M); zeros(M) R2];
alpha=0.5;
Kgpc=(G'*Q*G+R)\(G'*Q);
x01=0.0882;
x02=441.2;
%.....
dU1_=zeros(nb+d,length(t));
dU2_=zeros(nb+d,length(t));
dU_=[dU1_;dU2_];
d1=zeros(1,length(t));
%y1=0; %linear
y1=441.2;
u_1=[];
u_2=[];
ym=[];
y=0;
Y_d=zeros(P,length(t));
Y_past=zeros(P,length(t));
Y_m=zeros(P,length(t));
D=zeros(P,length(t));
E=zeros(P,length(t));
dU1=zeros(M,length(t));
dU2=zeros(M,length(t));

```

```

dU=[dU1;dU2];
U1=zeros(M,length(t));
U2=zeros(M,length(t));
Y_=zeros(na,length(t));
%.....step.....
r =ones(length(t),1);
%.....

for i=1:length(t)-1

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

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

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

dU(:,i+1)=Kgpc*E(:,i+1);
dU1(:,i+1)=dU(1:M,i+1);
dU2(:,i+1)=dU(M+1:2*M,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:nb+d,i+2) = dU1_(1:nb+d-1,i+1);
dU1_(1,i+2)=dU1(1,i+1);
dU2_(2:nb+d,i+2) = dU2_(1:nb+d-1,i+1);
dU2_(1,i+2)=dU2(1,i+1);
dU_(:,i+2)=[dU1_(:,i+2);dU2_(:,i+2)];
Y_(2:na,i+2)=Y_(1:na-1,i+1);
Y_(1,i+2)=Y_m(1,i+1);

u1=U1(1,i+1);
u2=U2(1,i+1);
sim('Model')
%d(i+2)=yl(end)-Y_m(1,i); %linear
d1(i+2)=y(end)-Y_m(1,i+1);
%y=yl(end); % linear
y=y(end);%+dist(i,1); % nonlinear
%y1=[y1;yl(end)]; % linear
y1=[y1; y+441.2];
ym=[ym; Y_m(1,i)];
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:150;
[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;
%.....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);
G=[G1 G2];
%.....
% A~=-1-2.564z^-1+2.2365z^-2-0.6725z^-3
% According to the discrete transfer function, below parameters have been
% defined
na=3;
nb1=1; nb2=1;

```

```

nb=nb1;
d=0;
N1=d+1;
N2=d+P;
a_=[1 -2.564 2.2365 -0.6725];
b1_=num1(2:end);
b2_=num2(2:end);
C=1; % because of using white noise
f=zeros(P+d,na+1);
f(1,1:3)=-1*a_(2:4);
for j=1:P+d-1
    for i=1:na
        f(j+1,i)=f(j,i+1)-f(j,1)*a_(i+1);
    end
end
F=f(N1:N2,1:na);
%.....
E1=zeros(P);
E1(:,1)=1;
for j=1:P-1
    E1(j+1:P,j+1)=f(j,1);
end
B1=zeros(P,P+nb);
for k=1:P
    B1(k,k:k+1)=b1_;
end
m1_=E1*B1;
M1_=zeros(P,nb+d);
for k=1:P
    M1_(k,:)=m1_(k,k+1);
end
%.....
E2=zeros(P);
E2(:,1)=1;
for j=1:P-1
    E2(j+1:P,j+1)=f(j,1);
end
B2=zeros(P,P+nb);
for k=1:P
    B2(k,k:k+1)=b2_;
end
m2_=E2*B2;
M2_=zeros(P,nb+d);
for k=1:P
    M2_(k,:)=m2_(k,k+1);
end
M_=[M1_ M2_];
%.....
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(P);
R1 =((1.2)^2)*gamma*gain_DC^2*eye(M);
R2=gamma*gain_DC2^2*eye(M);
R=[R1 zeros(M); zeros(M) R2];
alpha=0.5;

```



```

Kgpc=(G'*Q'*G+R)\(G'*Q);
x01=0.0882;
x02=441.2;
%.....
dU1_=zeros(nb+d,length(t));
dU2_=zeros(nb+d,length(t));
dU_=[dU1_;dU2_];
d1=zeros(1,length(t));
%y1=0; %linear
y1=441.2;
u_1=[];
u_2=[];
ym=[];
y=0;
Y_d=zeros(P,length(t));
Y_past=zeros(P,length(t));
Y_m=zeros(P,length(t));
D=zeros(P,length(t));
E=zeros(P,length(t));
dU1=zeros(M,length(t));
dU2=zeros(M,length(t));
dU=[dU1;dU2];
U1=zeros(M,length(t));
U2=zeros(M,length(t));
Y_=zeros(na,length(t));
%.....step.....
r =ones(length(t),1);
%.....

for i=1:length(t)-1

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

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

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

dU(:,i+1)=Kgpc*E(:,i+1);
dU1(:,i+1)=dU(1:M,i+1);
dU2(:,i+1)=dU(M+1:2*M,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:nb+d,i+2) = dU1_(1:nb+d-1,i+1);
dU1_(1,i+2)=dU1(1,i+1);
dU2_(2:nb+d,i+2) = dU2_(1:nb+d-1,i+1);
dU2_(1,i+2)=dU2(1,i+1);
dU_(:,i+2)=[dU1_(:,i+2);dU2_(:,i+2)];

```

```

Y_(2:na,i+2)=Y_(1:na-1,i+1);
Y_(1,i+2)=Y_m(1,i+1);

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

```

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]=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:10;
[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;
%.....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);
G=[G1 G2];
%.....
%  $A \sim 1 - 2.564z^{-1} + 2.2365z^{-2} - 0.6725z^{-3}$ 
% According to the discrete transfer function, below parameters have been
% defined
na=3;
nb1=1; nb2=1;
nb=nb1;
d=0;
N1=d+1;
N2=d+P;
a_=[1 -2.564 2.2365 -0.6725];
b1_=num1(2:end);
b2_=num2(2:end);
C=1; % because of using white noise
f=zeros(P+d,na+1);
f(1,1:3)=-1*a_(2:4);
for j=1:P+d-1
    for i=1:na
        f(j+1,i)=f(j,i+1)-f(j,1)*a_(i+1);
    end
end
F=f(N1:N2,1:na);
%.....
E1=zeros(P);
E1(:,1)=1;
for j=1:P-1
    E1(j+1:P,j+1)=f(j,1);
end
B1=zeros(P,P+nb);
for k=1:P
    B1(k,k:k+1)=b1_;
end
m1_=E1*B1;

```

```

M1_=zeros(P,nb+d);
for k=1:P
    M1_(k,:)=m1_(k,k+1);
end
%.....
E2=zeros(P);
E2(:,1)=1;
for j=1:P-1
    E2(j+1:P,j+1)=f(j,1);
end
B2=zeros(P,P+nb);
for k=1:P
    B2(k,k:k+1)=b2_;
end
m2_=E2*B2;
M2_=zeros(P,nb+d);
for k=1:P
    M2_(k,:)=m2_(k,k+1);
end
M_=[M1_ M2_];
%.....
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(P);
R1 =((1.2)^2)*gamma*gain_DC^2*eye(M);
R2=gamma*gain_DC2^2*eye(M);
R=[R1 zeros(M); zeros(M) R2];
alpha=0.5;
Kgpc=(G'*Q*(G+R))/(G'*Q);
x01=0.0882;
x02=441.2;
%.....
dU1_=zeros(nb+d,length(t));
dU2_=zeros(nb+d,length(t));
dU_=[dU1_;dU2_];
d1=zeros(1,length(t));
%y1=0; %linear
y1=441.2;
u_1=[];
u_2=[];
ym=[];
y=0;
Y_d=zeros(P,length(t));
Y_past=zeros(P,length(t));
Y_m=zeros(P,length(t));
D=zeros(P,length(t));
E=zeros(P,length(t));
dU1=zeros(M,length(t));
dU2=zeros(M,length(t));
dU=[dU1;dU2];
U1=zeros(M,length(t));
U2=zeros(M,length(t));
Y_=zeros(na,length(t));
% dist=zeros(length(t),1);
% dist(18:26,1)=ones(9,1);

```



```

dU1(:,i+1)=dU(1:M,i+1);
dU2(:,i+1)=dU(M+1:2*M,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:nb+d,i+2) = dU1_(1:nb+d-1,i+1);
dU1_(1,i+2)=dU1(1,i+1);
dU2_(2:nb+d,i+2) = dU2_(1:nb+d-1,i+1);
dU2_(1,i+2)=dU2(1,i+1);
dU_(:,i+2)=[dU1_(:,i+2);dU2_(:,i+2)];
Y_(2:na,i+2)=Y_(1:na-1,i+1);
Y_(1,i+2)=Y_m(1,i+1);

u1=U1(1,i+1);
u2=U2(1,i+1);
sim('Model')
%d(i+2)=yl(end)-Y_m(1,i); %linear
d1(i+2)=y(end)-Y_m(1,i+1);
%y=yl(end); % linear
y=y(end);%+dist(i,1); % nonlinear
%y1=[y1;yl(end)]; % linear
y1=[y1; y+441.2];
ym=[ym; Y_m(1,i)];
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);
plot(y1,'b');
hold on
plot(r+441.2,'r');
grid on
%axis([0 600 440 448]);
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])

```

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

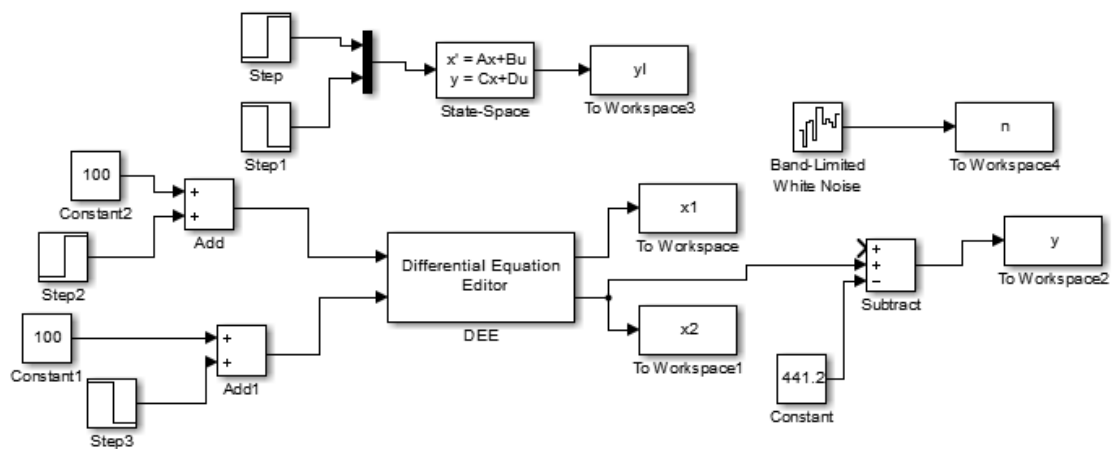


Figure 17. Simulation of the linear and nonlinear plant in Simulink.