



7.2.1395

پروژه دوم درس MPC

طراحی کنترل کننده DMC

نام دانشجو: فاطمه زاهدی

شماره دانشجویی: 94201583

نام استاد: دکتر حائری

فهرست:

1. معرفی سیستم مورد استفاده: 2
2. طراحی کنترل کننده DMC: 2
3. پاسخ سیستم به ازای خروجی های مطلوب مختلف: 4
- 3.1. خروجی مطلوب پالس: 4
- 3.2. خروجی مطلوب سینوس: 6
- 3.3. خروجی مطلوب پله ای با پرش های مختلف: 7
- 3.4. خروجی مطلوب پله ای: 9
4. بررسی اثر پارامترها: 14
- 4.1. تغییر ماتریس R: 14
- 4.2. تغییر ماتریس Q: 15
- 4.3. تغییر مقدار α : 16
- 4.4. تغییر مقدار N: 18
- 4.5. تغییر مقدار P یا M: 19
- 4.6. تغییر مقدار T_s : 20
5. بررسی اثر اغتشاش و نویز: 22
- 5.1. اثر اغتشاش: 22
- 5.2. اثر نویز: 24
6. بررسی عدم قطعیت در پروسه: 26
7. کدها و شبیه سازی ها: 28

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.

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

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

2. طراحی کنترل کننده DMC:

در این طراحی ابتدا بایستی به تعیین پارامترهای T_s , P , N 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 $M = P = t_r/T_s$ بدست آوردیم. 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. مقدار N نیز به ازای هر دو ورودی با استفاده از رابطه‌ی $N = t_s/T_s$ بدست آوردیم و مقدار بزرگتر به عنوان N برای طراحی کنترلر استفاده شده که این مقدار برابر با 17 شد.

بعد از بدست آوردن مقدار پارامترهای مذکور بایستی به تشکیل ماتریس toeplitz و hankel به ازای هر کدام از ورودی‌ها بپردازیم اما در این پروژه g_i ها را با استفاده از سیستم گسسته شده پس از خطی سازی به دست نمی آوریم بلکه با استفاده از خود سیستم غیر خطی این مقادیر را بدست آورده و ماتریس‌های مذکور را تشکیل می دهیم. این کار را بدین روش انجام می دهیم که 4 پله به سیستم غیر خطی حول نقطه کار می دهیم و g_i ها را با استفاده از خروجی سیستم غیر خطی بدست می آوریم.

نحوه تشکیل ماتریس toeplitz و hankel در کنترل کننده DMC به صورت زیر می باشد.

$$G = \begin{pmatrix} g_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ g_p & \cdots & g_1 \end{pmatrix}, \quad G^- = \begin{pmatrix} g_2 & \cdots & g_N \\ \vdots & \ddots & \vdots \\ g_{p+1} & \cdots & 0 \end{pmatrix}, \quad G = [G_1 \ G_2], \quad G^- = [G_1^- \ G_2^-]$$

بردار ΔU مربوط به تغییرات ورودی‌های آینده سیستم و بردار ΔU^- مربوط به تغییرات ورودی‌های گذشته سیستم می باشد. این دو بردار در زیر آورده شده اند.

$$\Delta U = \begin{pmatrix} \Delta u_1(t) \\ \vdots \\ \Delta u_1(t+P-1) \\ \Delta u_2(t) \\ \vdots \\ \Delta u_2(t+P-1) \end{pmatrix}, \quad \Delta U^- = \begin{pmatrix} \Delta u_1(t-1) \\ \vdots \\ \Delta u_1(t-N+1) \\ \Delta u_2(t-1) \\ \vdots \\ \Delta u_2(t-N+1) \end{pmatrix}, \quad U^- = \begin{pmatrix} u(t-N) \\ \vdots \\ u(t-N+P-1) \end{pmatrix}.$$

Y_m که خروجی مدل است به صورتی که در آورده شده است می‌باشد.

$$Y_m = GU + Y_{past}, \quad Y_{past} = G^- \Delta U^- + g1_{N+1} U1^- + g2_{N+1} U2^-$$

هدف بدست آوردن تغییرات ورودی‌های آینده ΔU به گونه ای است که تابع هدف زیر مینیمم گردد و خروجی تا حد امکان خروجی مطلوب را دنبال نماید.

$$J = (Y_p - Y_d)^T Q (Y_p - Y_d) + \Delta U^T R \Delta U$$

در اینجا Y_p خروجی پروسه که از سیستم غیر خطی بدست می‌آید و Y_d خروجی مطلوب می‌باشد.

با توجه به اینکه سیستم را در حالت حلقه بسته بررسی می‌کنیم D مخالف صفر است و از رابطه زیر بدست می‌آید.

$$Y_p = Y_m + D$$

به منظور انتخاب Q و R بدین صورت عمل میکنیم که میخواهیم وزن این دو در تابع هدف تقریباً یکسان باشد. بنابراین Q را برابر با I قرار می‌دهیم. از طرفی چون دو ورودی داریم به ازای هر ورودی R_1 و R_2 را بدست می‌آوریم و روی قطر ماتریس R قرار می‌دهیم. ماتریس R به صورت زیر بدست می‌آید.

$$R_1 = \gamma (gain_{DC1})^2 r I, \quad R_2 = \gamma (gain_{DC2})^2 I, \quad R = diag([R_1 \ R_2]).$$

که این r به منظور هم وزن کردن R های ورودی‌ها با هم است زیرا به خاطر گینشان ممکن است تفاوت وزن ایجاد شود که در این سیستم ما این r را $(1.2)^2$ گرفته ایم.

با استفاده از روابط بالا کنترل کننده DMC را به صورت programmed طراحی کرده و بررسی این کنترل کننده از نظر تغییرات پارامترها، حضور نویز و اغتشاش در خروجی پروسه و بررسی عدم قطعیت‌ها در بخش‌های بعد آورده شده است.

در این بخش پاسخ سیستم به ازای ورودی پله در حالی که گاما یک و آلفا 0.5 است بررسی شده است و نتیجه در شکل زیر آورده شده است.

همانطور که در شکل زیر می‌بینیم با وجود اینکه گاما مخالف صفر است خطای حالت ماندگار صفر می‌باشد. این به علت این است که سیستم ما تایپ 0 است و کنترل کننده DMC خطای حالت ماندگار را به صفر می‌رساند.

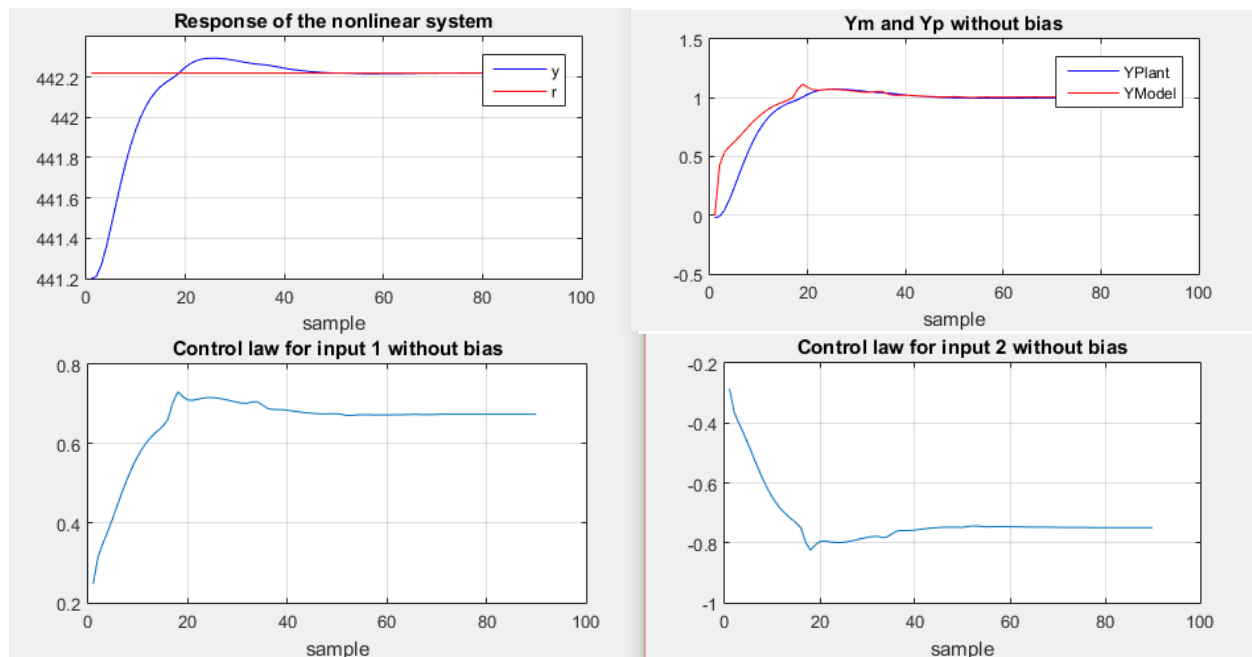


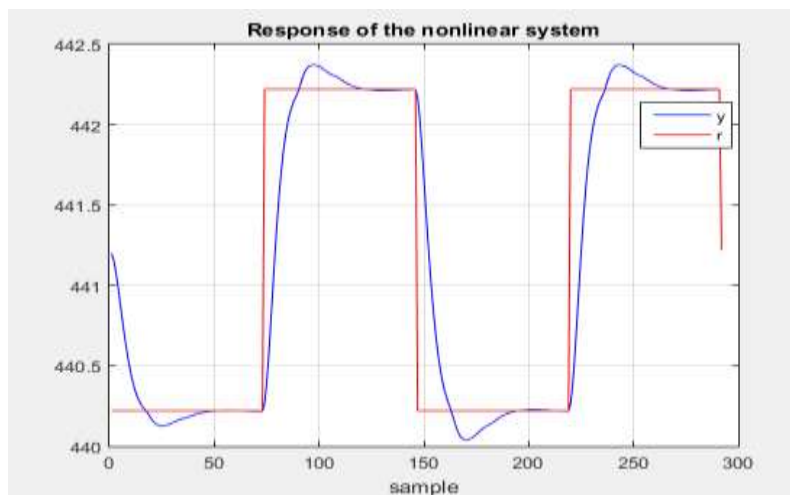
Figure1. DMC on nonlinear system.

3. پاسخ سیستم به ازای خروجی های مطلوب 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.



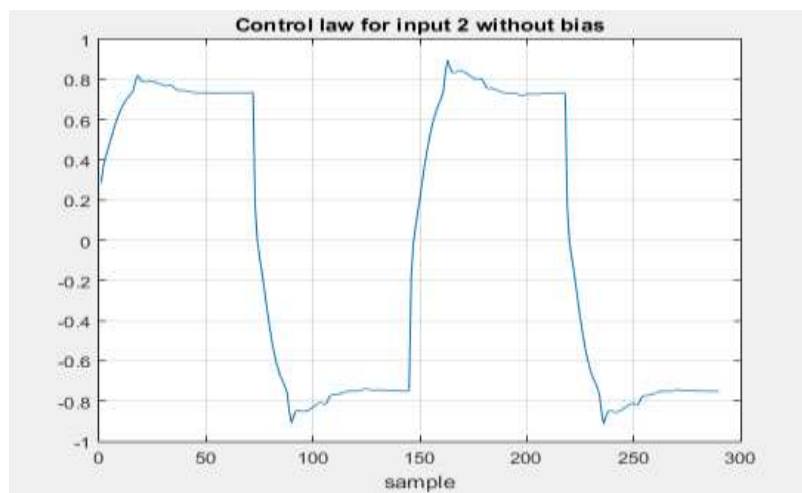
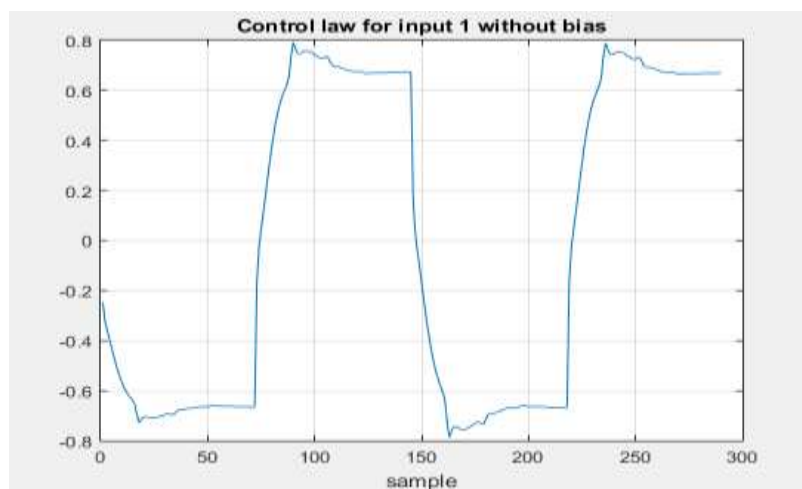
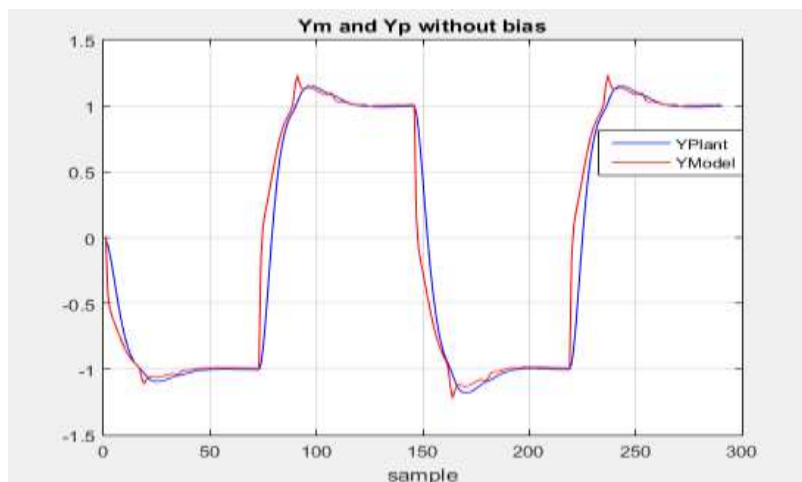
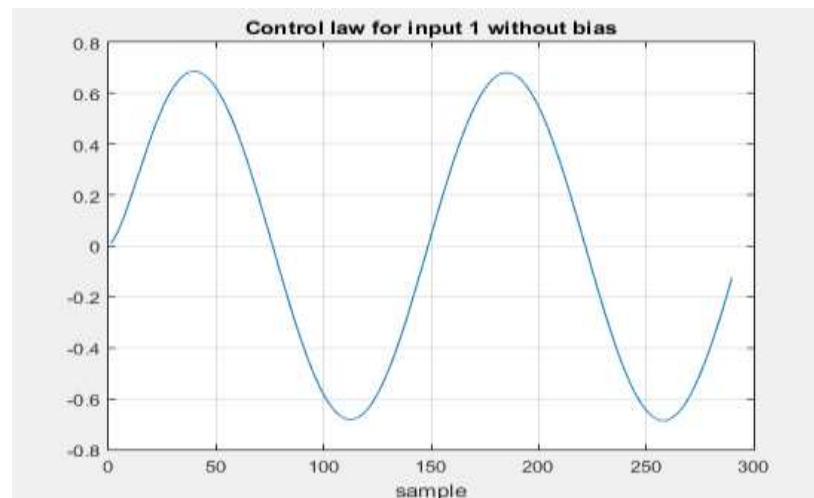
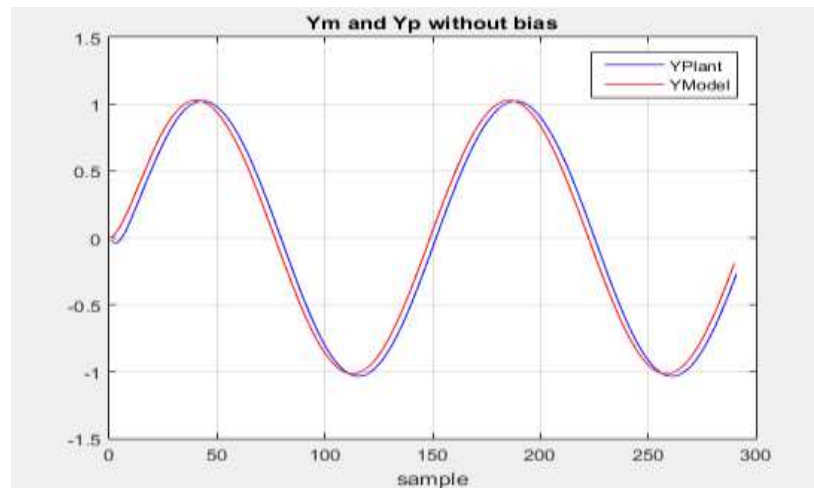
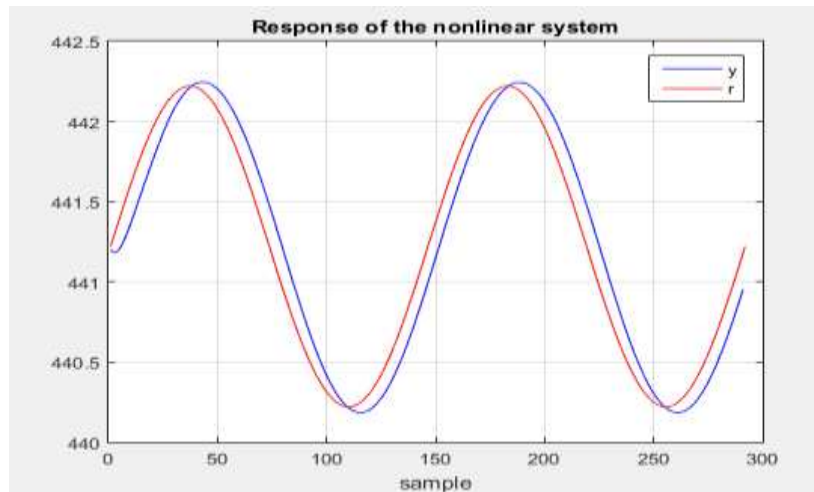


Figure 2. Response of the system with pulse as a desired output.

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

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.



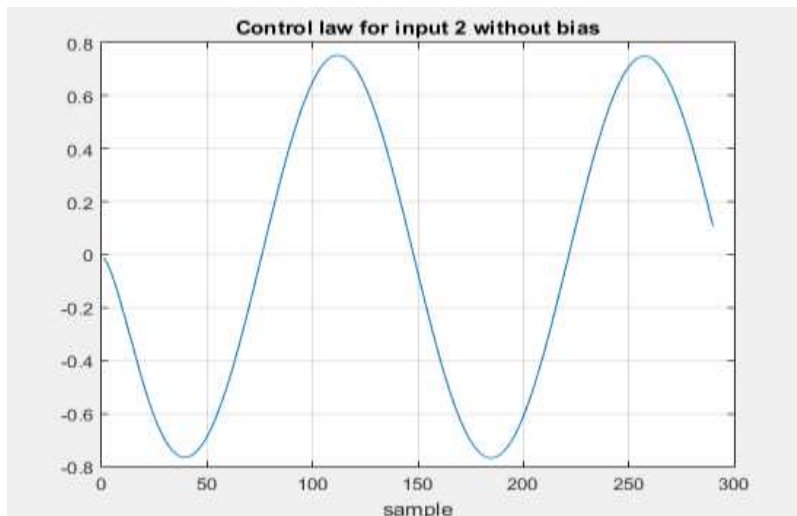
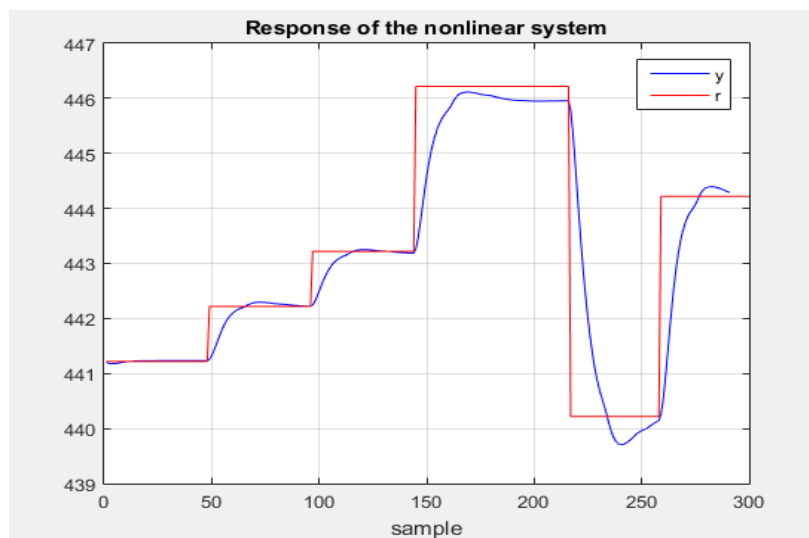


Figure 3. Response of the system with sine as a desired output.

همانطور که در شکل بالا می بینیم خروجی سیستم، خروجی مطلوب را دنبال می کند. قوانین کنترلی و خروجی مدل نیز در این شکل قرار داده شده است و می توان نحوه رفتار آنها را دید.

3.3. خروجی مطلوب پله ای با پرش های مختلف:

در این قسمت به ازای خروجی مطلوب سیستم، استپ با پرش های متفاوت می دهیم و پاسخ سیستم به ازای این خروجی مطلوب به صورت زیر شد. همانطور که در شکل زیر می بینیم خروجی سیستم، خروجی مطلوب را دنبال می کند اما در قسمت هایی که پرش شدیدتری اتفاق افتاده تا حدودی خطا ایجاد شده است.



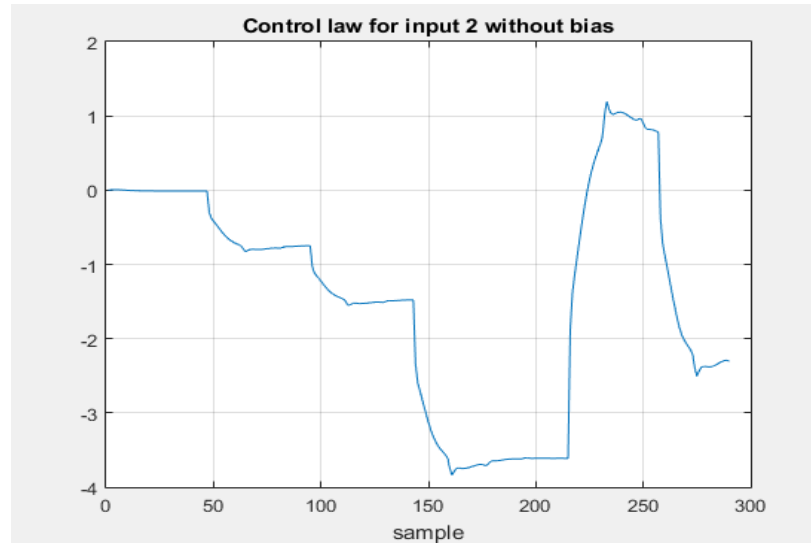
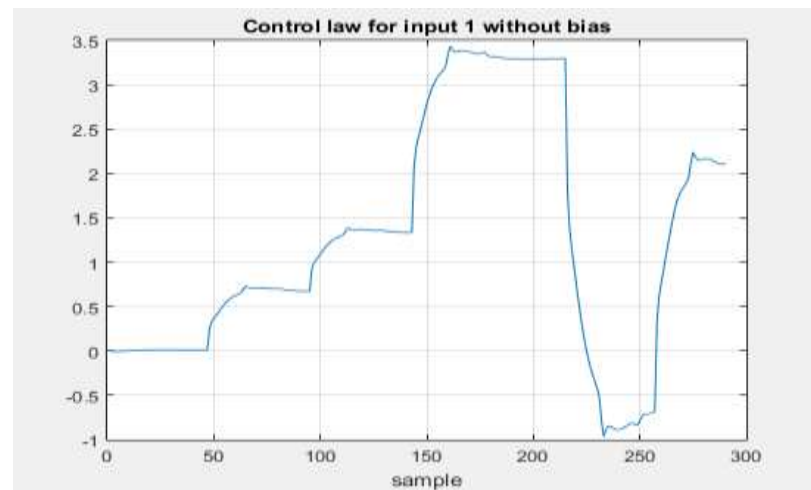
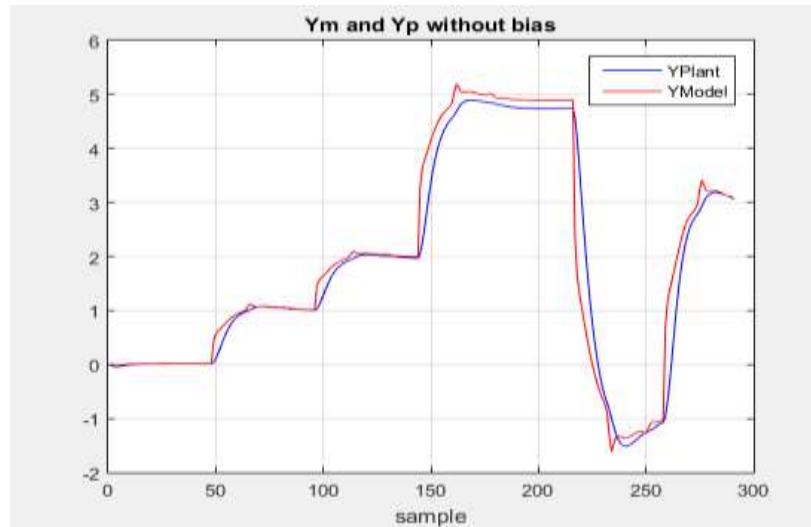
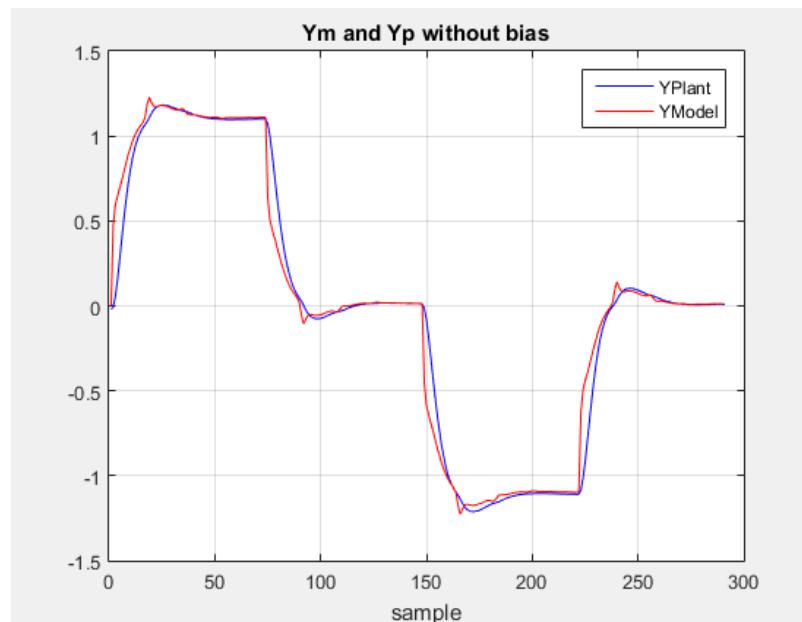
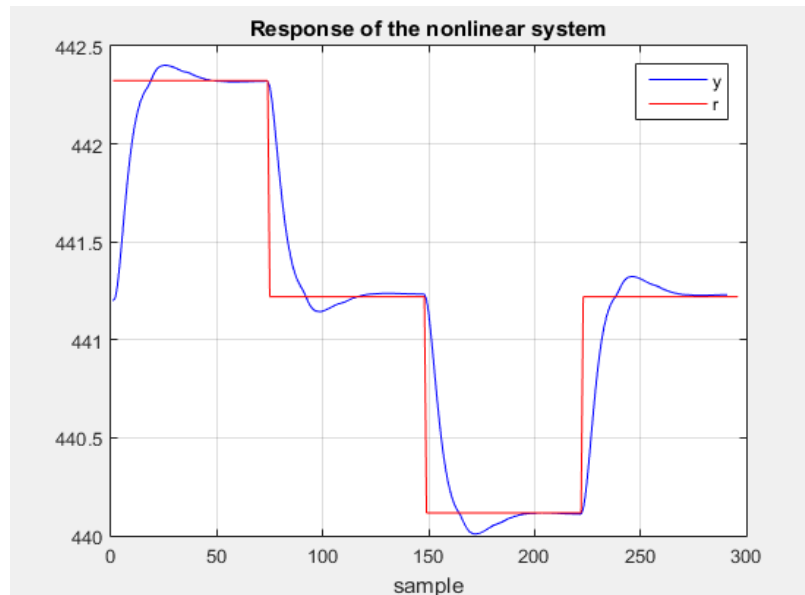


Figure 4. Response of the system with step with various jumps as a desired output.

3.4. خروجی مطلوب پله‌ای:

در این قسمت پاسخ سیستم به ازای خروجی مطلوب سیگنال پله ای را رسم می کنیم. این خروجی مطلوب را به ازای دامنه های مختلف شامل 0.25%، 1% و 2% نقطه کار رسم می کنیم. شکل زیر پاسخ را به ازای این خروجی مطلوب با دامنه 0.25% نقطه کار نشان می دهد.



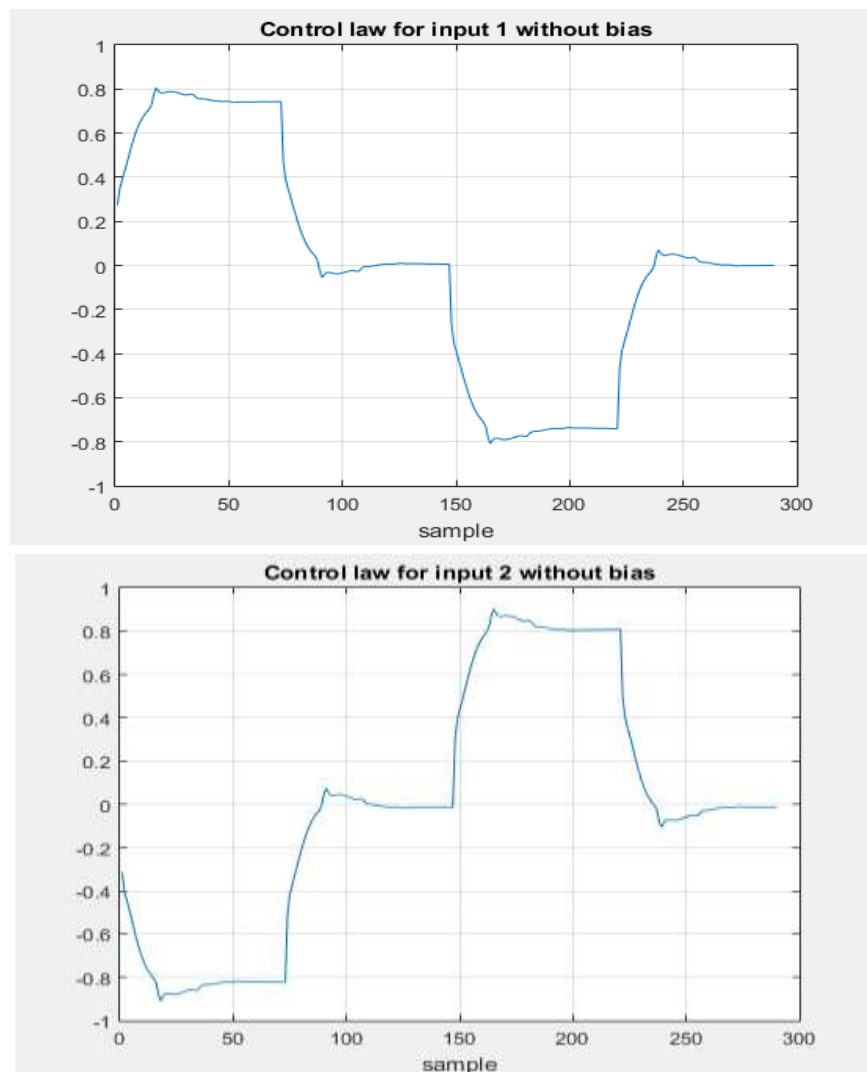
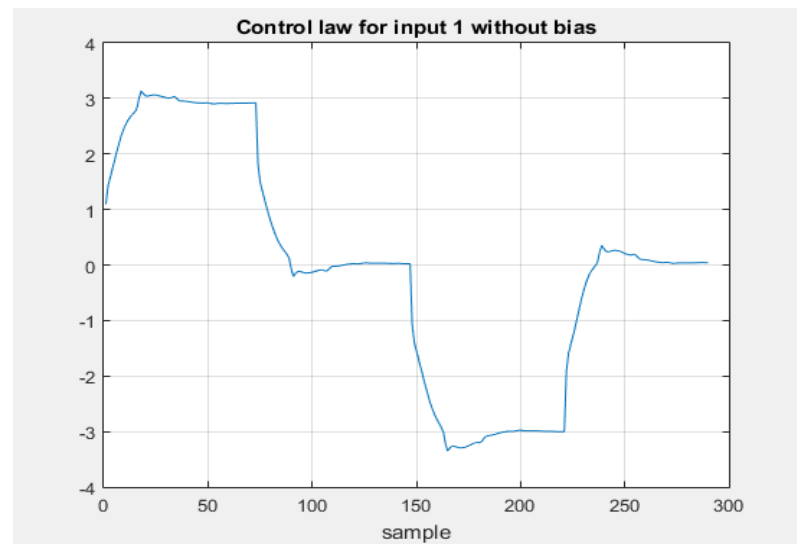
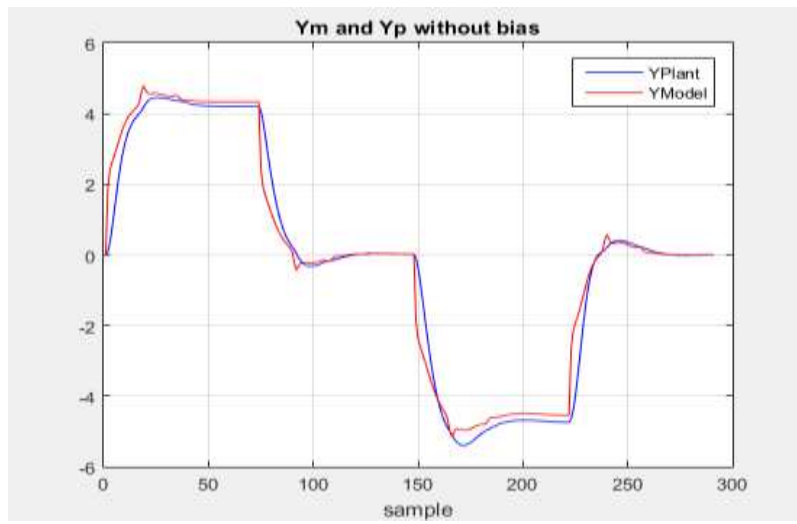
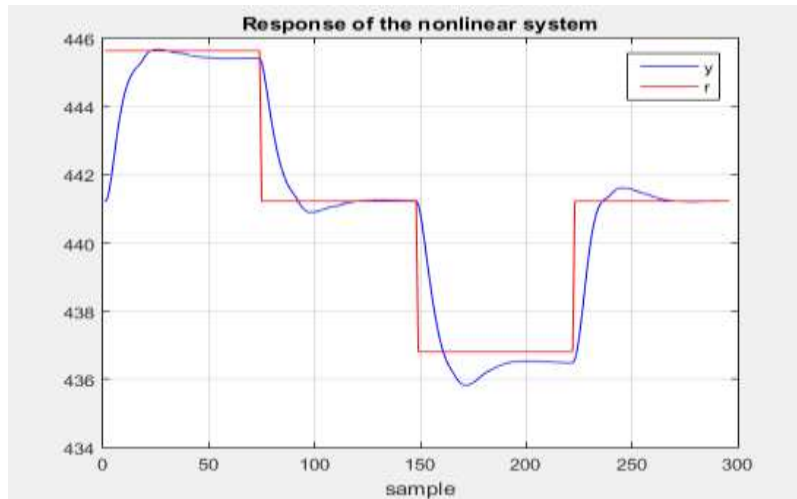


Figure 5. Response of the system with step with 0.25% of operating point as a desired output.

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

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

همانطور که در شکل زیر می‌بینیم پاسخ سیستم، خروجی مطلوب را دنبال می‌کند و در ناحیه‌هایی که خروجی مطلوب صفر است پاسخ سیستم نیز صفر است. اما می‌بینیم که در ناحیه‌هایی که خروجی مطلوب صفر نیست مقداری خطا نسبت به حالت قبل که دامنه خروجی مطلوب درصد کمتری از نقطه کار را داشت، ایجاد شده است و اثر غیر خطی بودن سیستم را با افزایش درصد دامنه خروجی مطلوب می‌توانیم ببینیم.



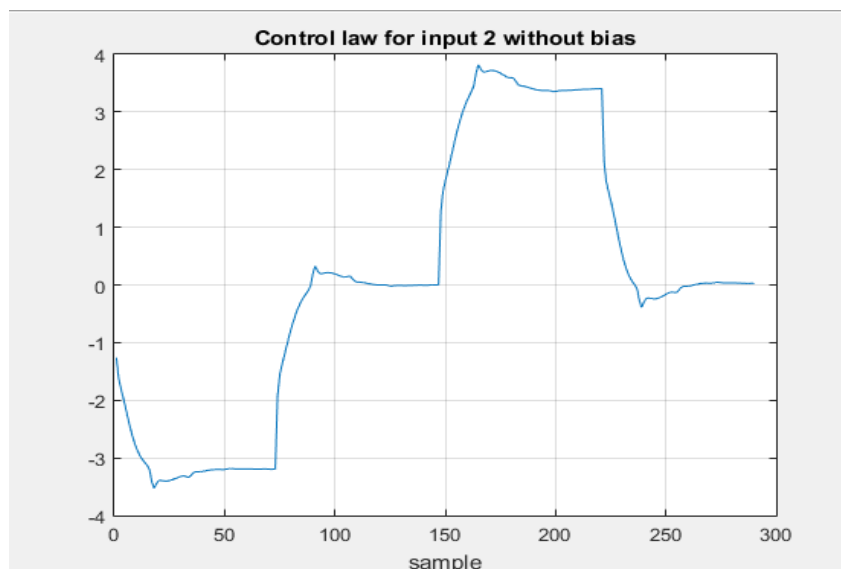
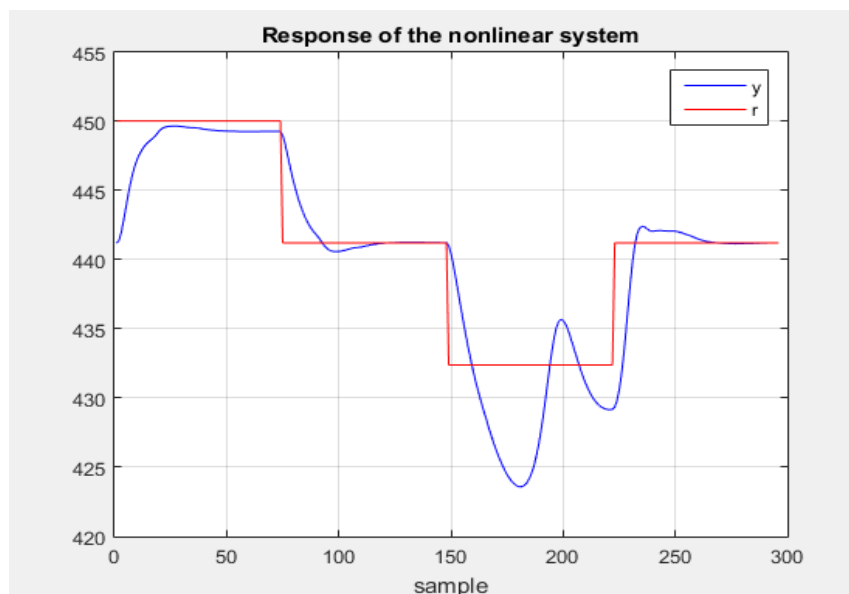


Figure 6. Response of the system with step with 1% of operating point as a desired output.

در شکل زیر پاسخ سیستم به ازای همین خروجی مطلوب اما به ازای دامنه 2% نقطه کار می‌بینیم. همانطور که در شکل 7 می‌بینیم باز هم در ناحیه‌هایی که خروجی مطلوب صفر است همچنان پاسخ سیستم صفر مانده است اما خطا در ناحیه‌هایی که خروجی مطلوب صفر نیست نسبت به حالت قبل افزایش یافته‌است. بنابراین تاثیر غیر خطی بودن سیستم را می‌توان با افزایش دامنه ورودی دید.



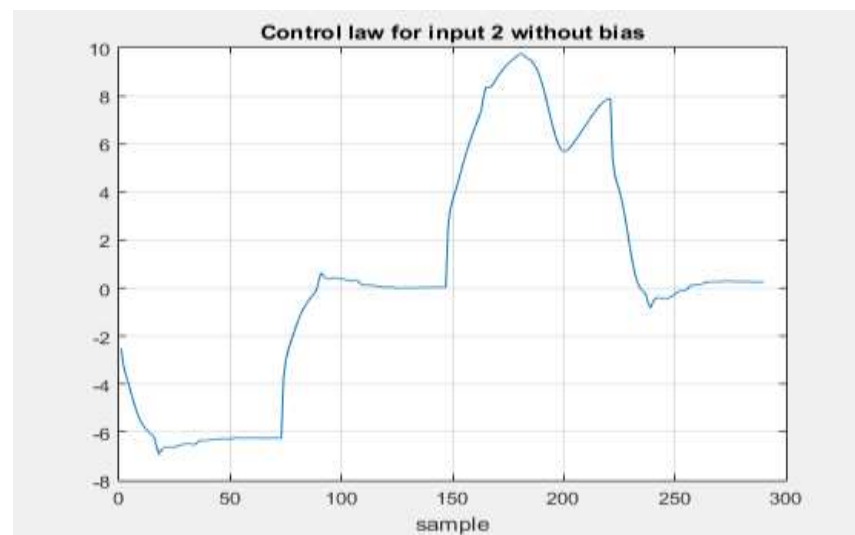
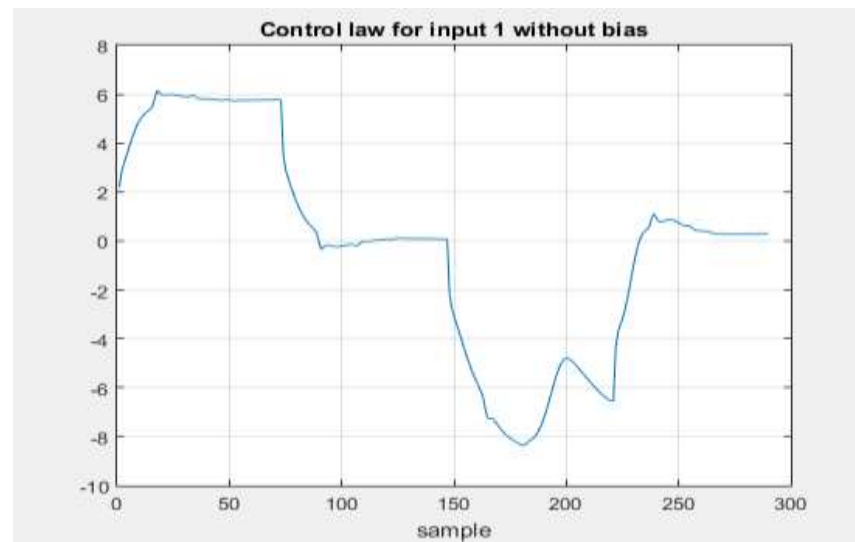
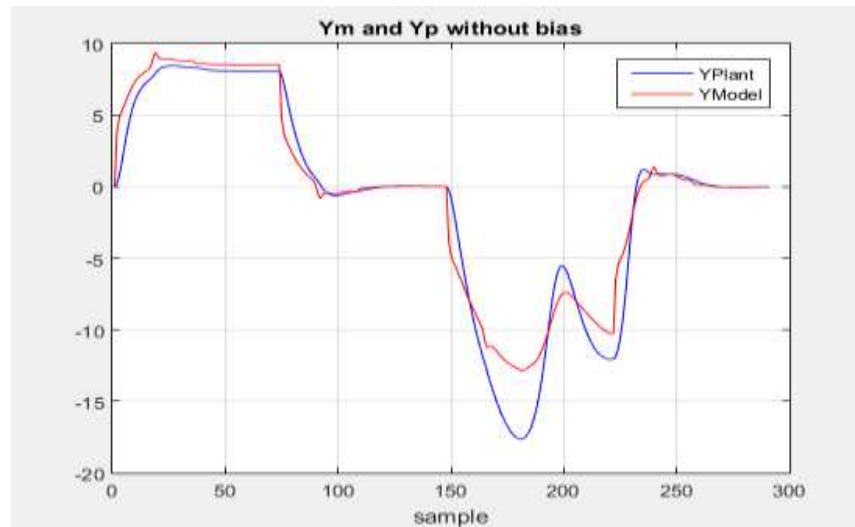


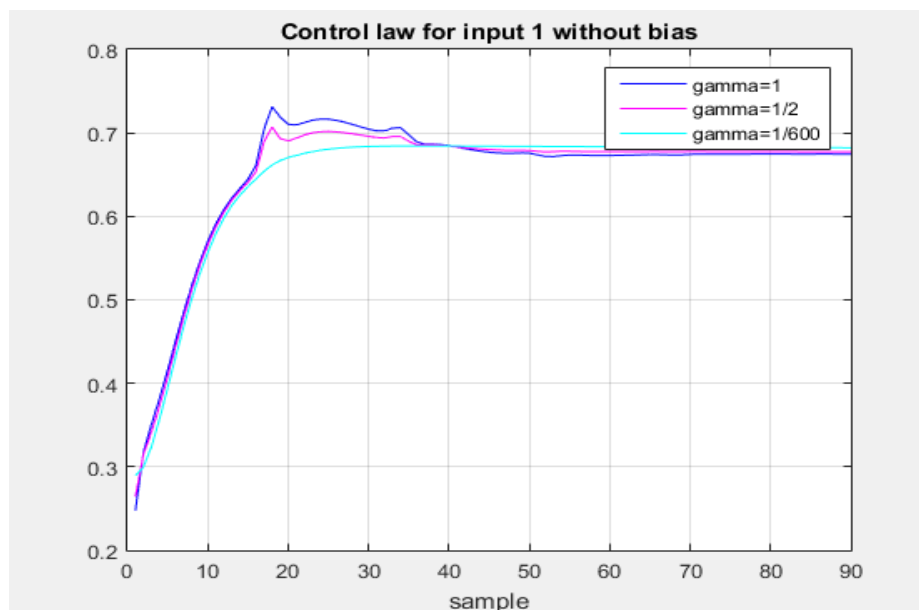
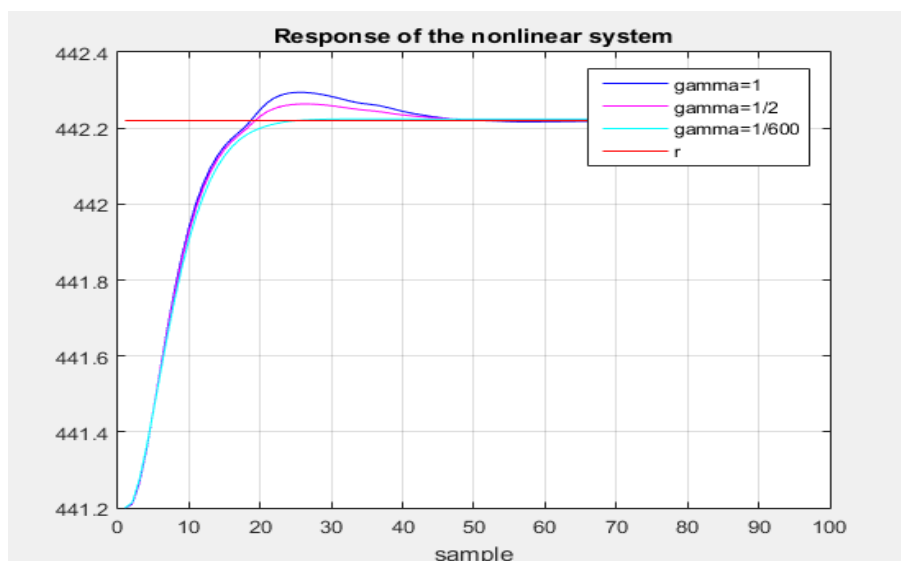
Figure 7. Response of the system with step with 2% of operating point as a desired output.

4. بررسی اثر پارامترها:

در این بخش تاثیر تغییرات پارامترهای مورد استفاده در طراحی کنترلر DMC را بر روی پاسخ سیستم بررسی می کنیم.

4.1. تغییر 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 می کنیم 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 می دهد.



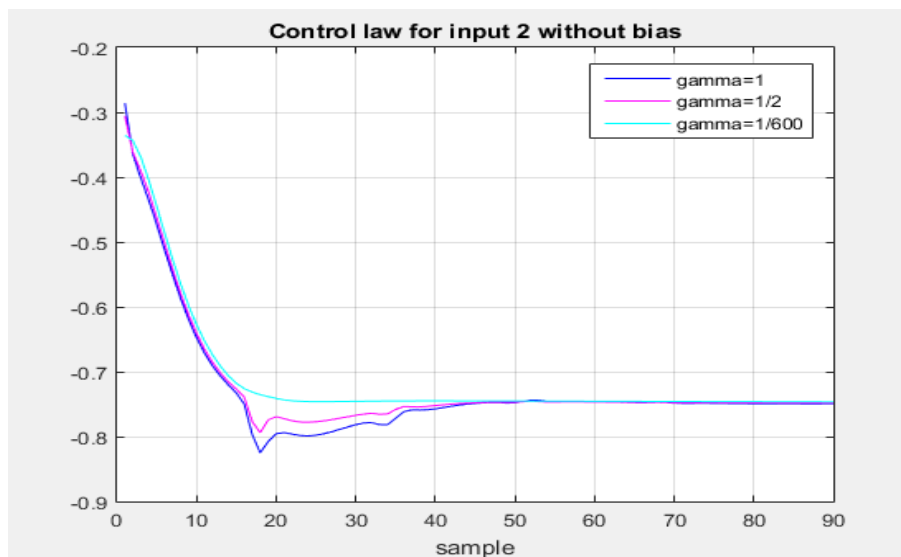


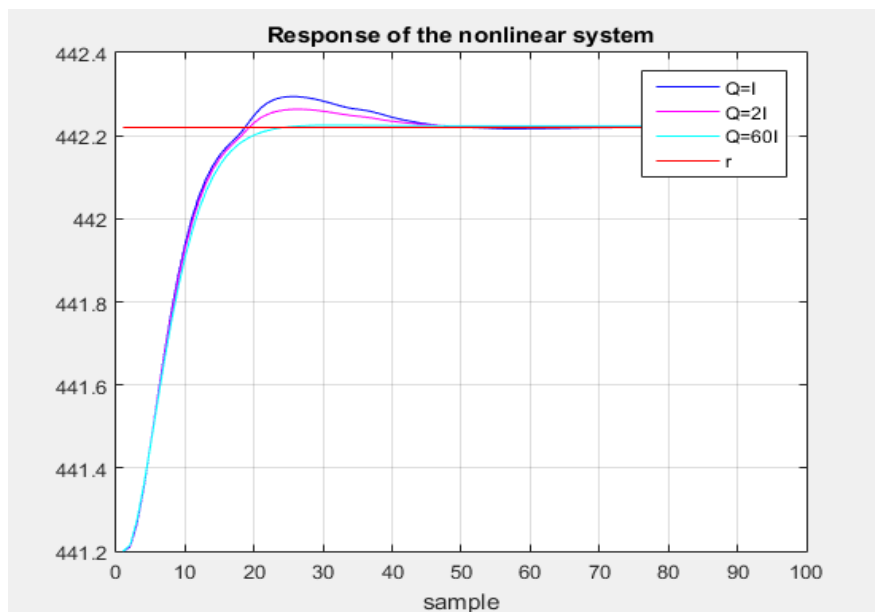
Figure 8. Responses of the system in verifying R .

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. تغییر 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 می‌کنیم.

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



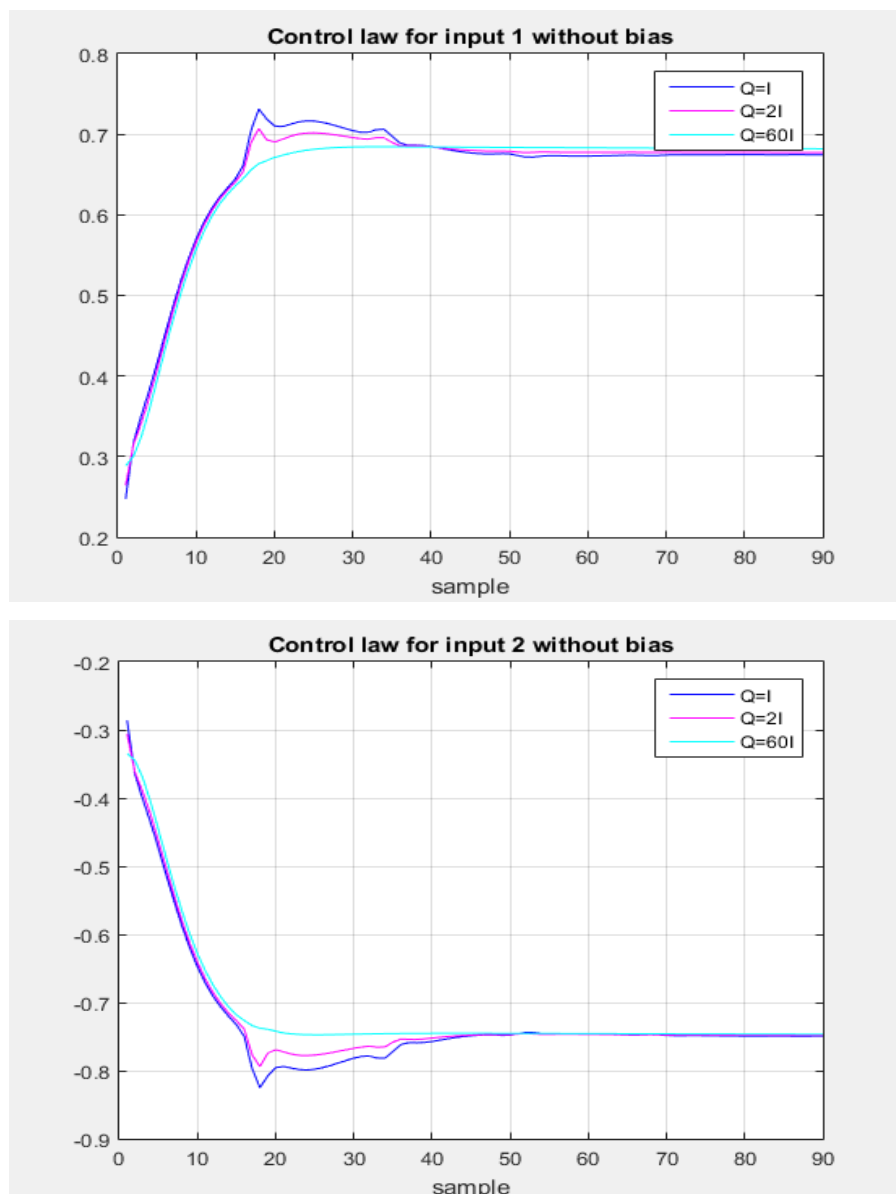


Figure 9. Responses of the system in verifying Q .

همانطور که در شکل 9 می‌توان دید با افزایش Q پاسخ سیستم تندتر می‌شود. اما دقیقاً مشابه کاهش γ ، خطای حالت ماندگار تغییری نمی‌کند و همواره صفر می‌ماند.

4.3. تغییر مقدار α :

در این قسمت تاثیر پارامتر α را بر روی پاسخ سیستم بررسی می‌کنیم. شکل زیر پاسخ سیستم به ازای مقادیر α 0.1، 0.4 و 0.6 نشان می‌دهد. در طی این بررسی مقدار γ یک قرار داده شده است.

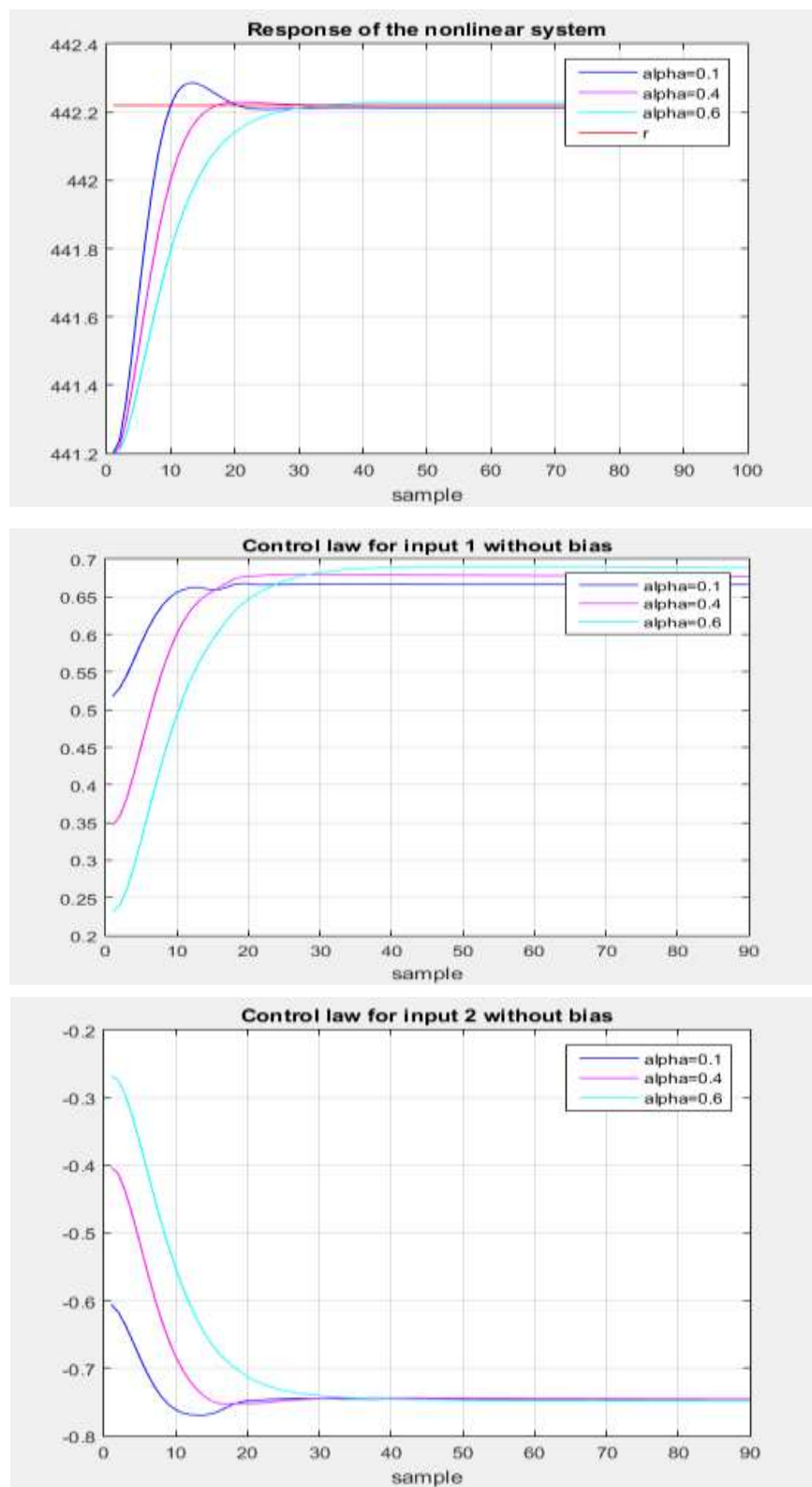
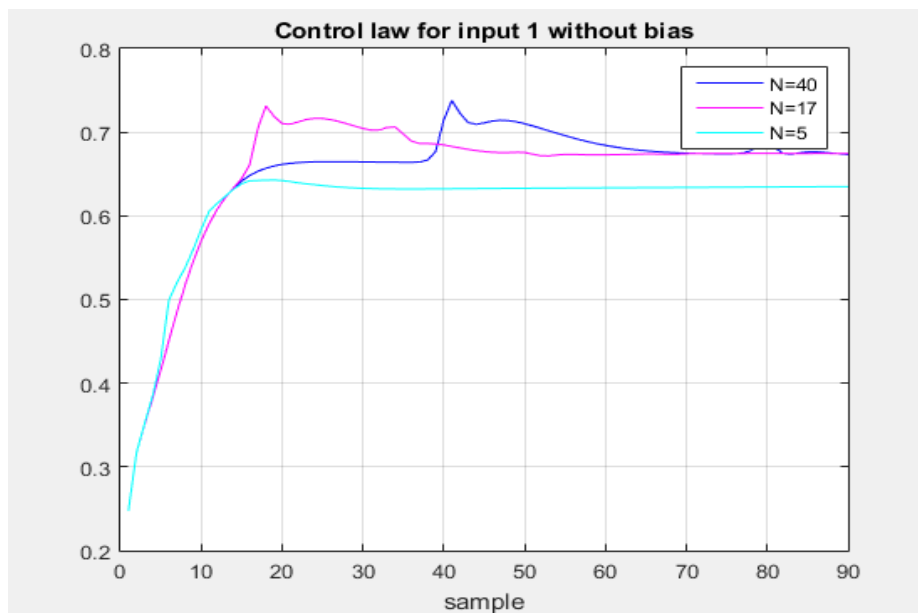
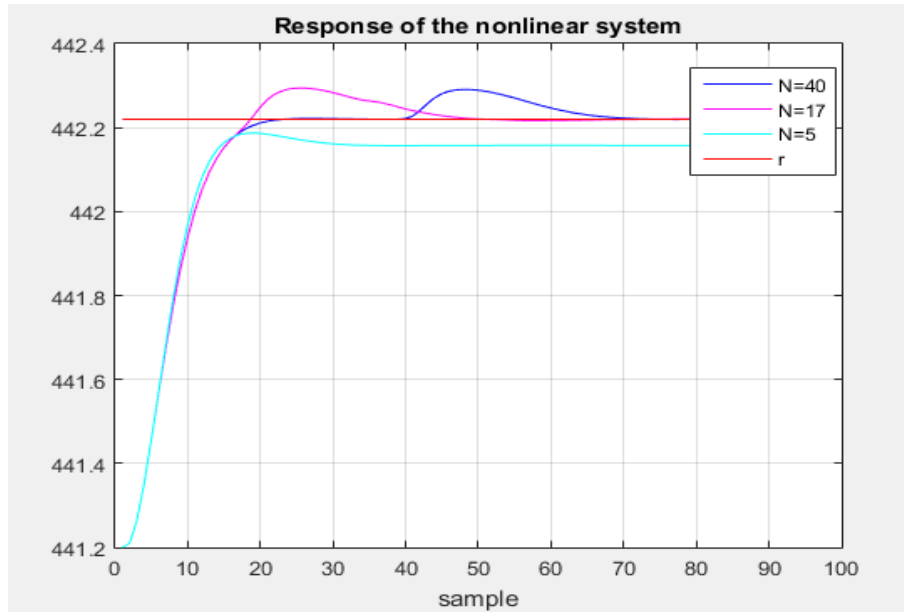


Figure 10. Responses of the system in verifying α .

همانطور که در شکل بالا می بینیم کاهش α باعث کند شدن پاسخ سیستم شده است.

4.4. تغییر مقدار N :

در این قسمت پاسخ سیستم را به ازای سه N مختلف شامل 5، 17 و 40 بررسی می کنیم. لازم به ذکر است که در این بررسی گاما برابر با یک و آلفا برابر با 0.5 قرار داده شده است. شکل زیر پاسخ سیستم به ازای این سه N نشان می دهد.



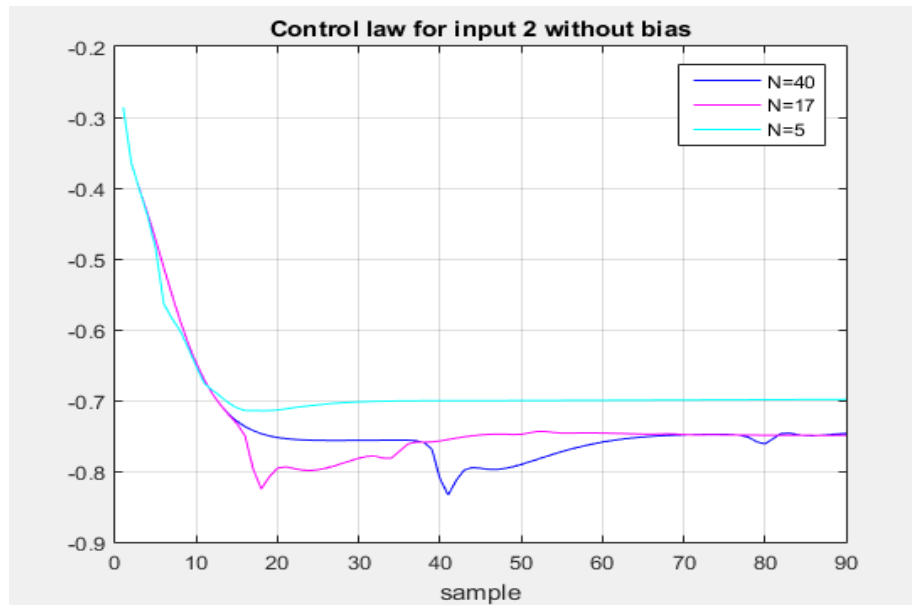
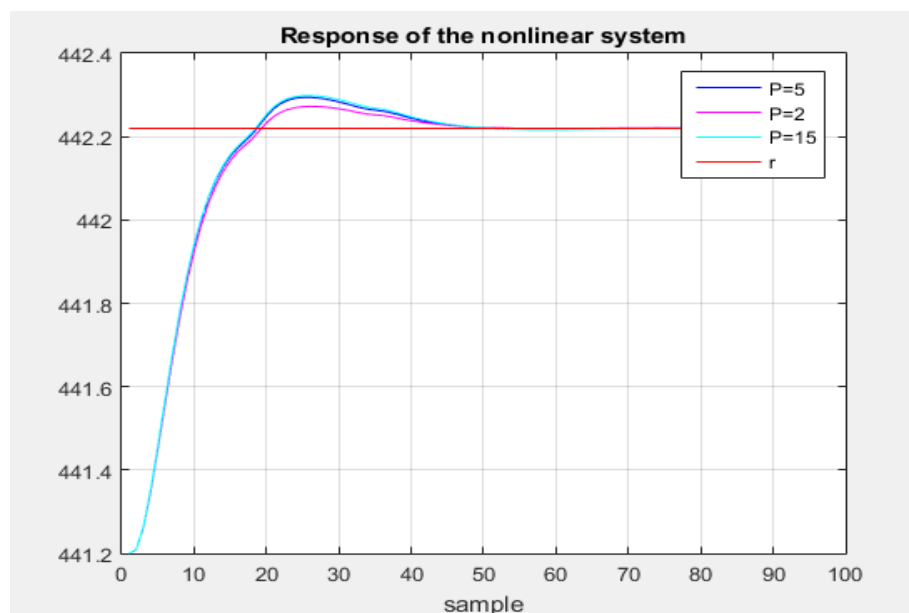


Figure 11. Responses of the system in verifying N .

همانطور که در شکل 11 می‌بینیم با کم شدن مقدار N تعداد نمونه‌ها کاهش یافته و این موضوع باعث شده که اطلاعات را از دست بدهیم. این موضوع را می‌توان در این شکل دید که به ازای $N=5$ چون تعداد نمونه‌ها کم است نتوانسته هنوز به مقدار بایاس صفر برسد.

4.5. تغییر مقدار 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.



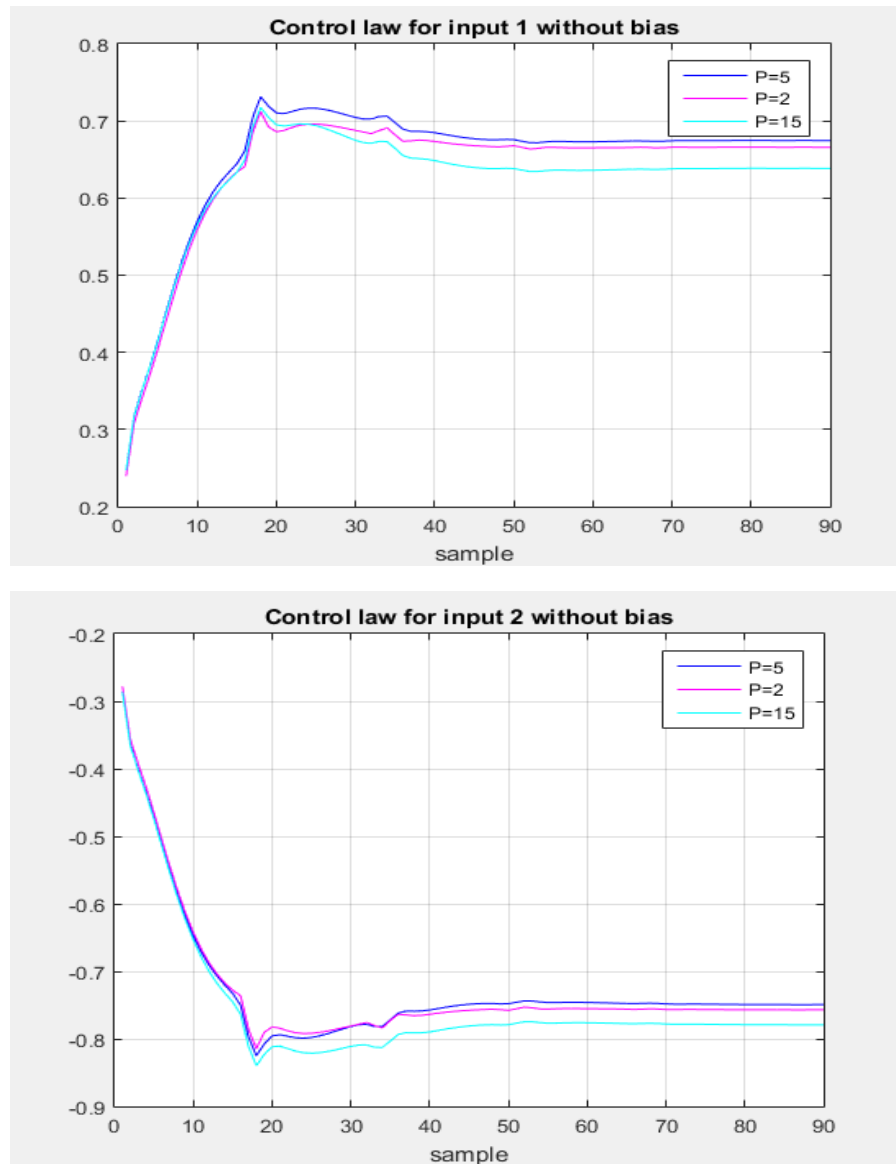


Figure 12. Responses of the system in verifying P or M .

همانطور که در شکل 12 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.

4.6. تغییر مقدار 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.

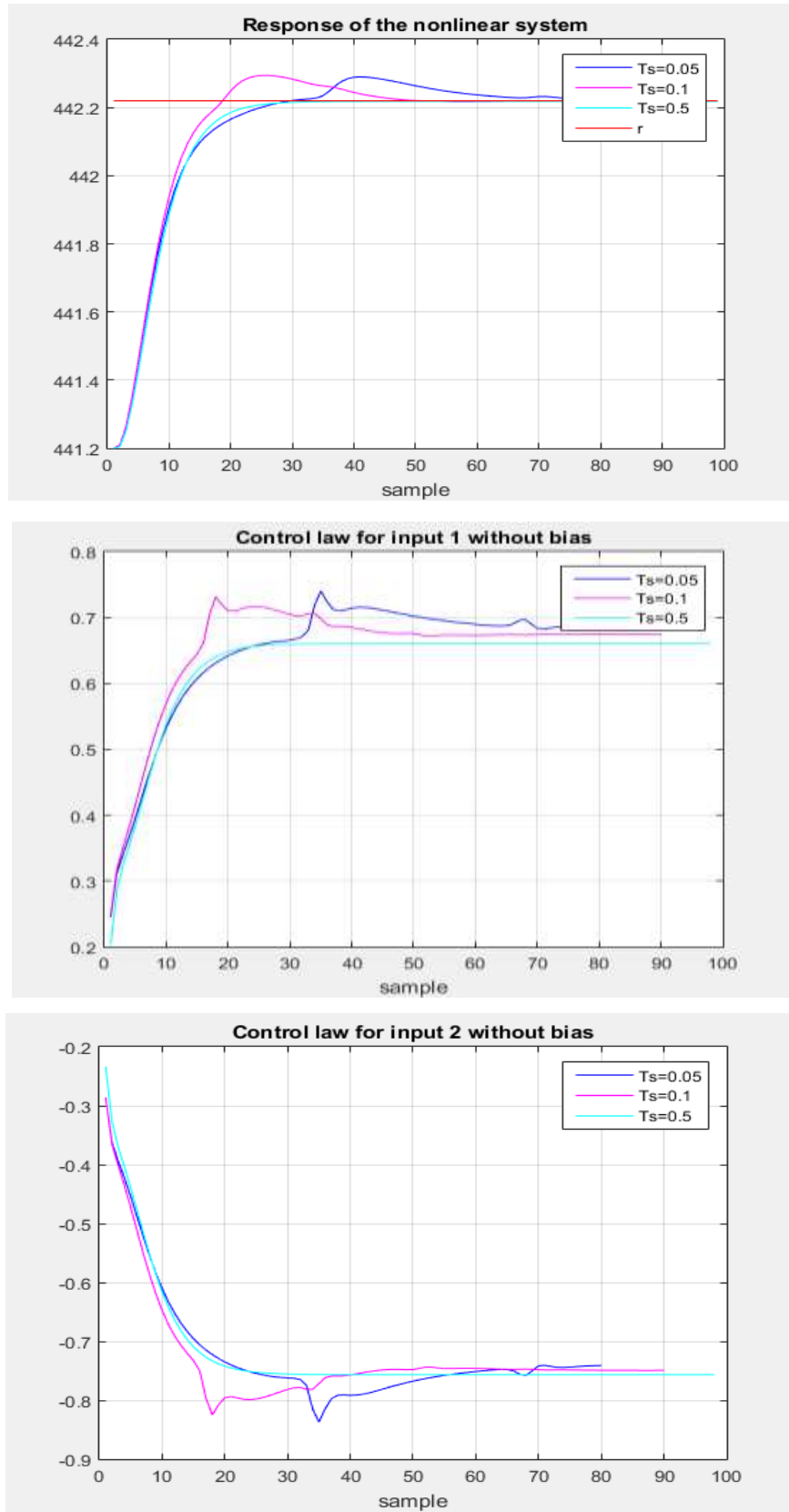


Figure 13. Responses of the system in verifying T_s .

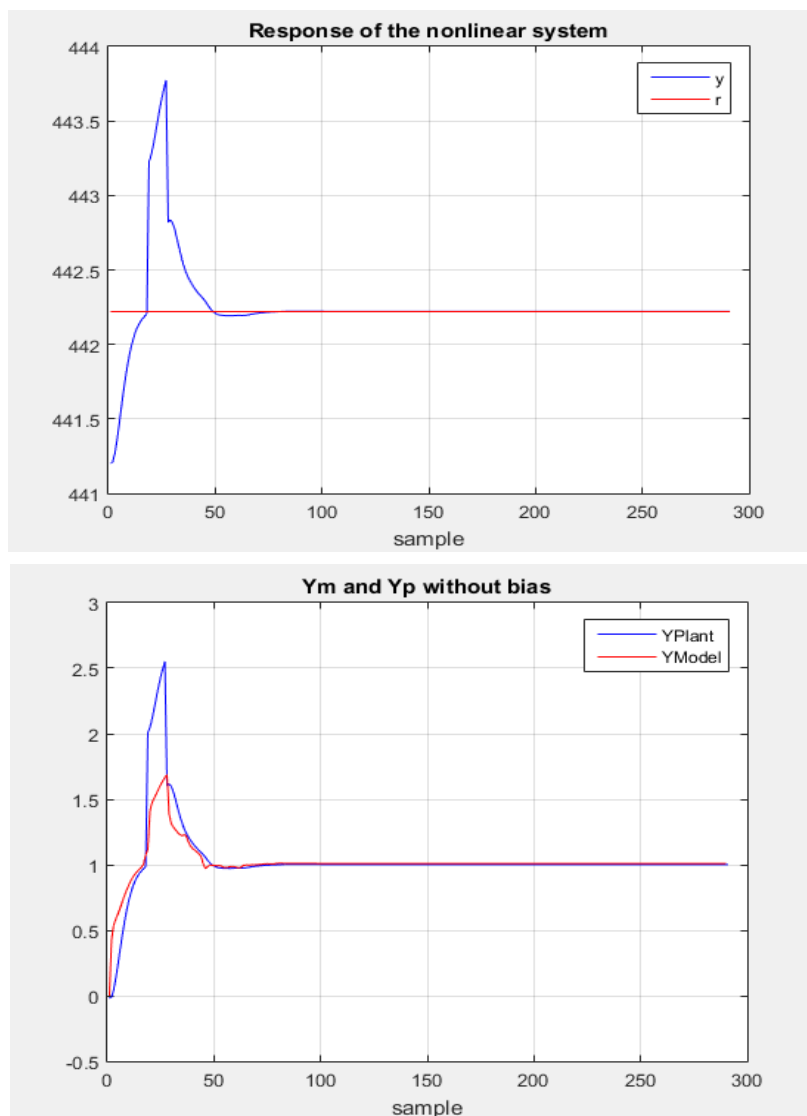
همانطور که در شکل 13 می‌بینیم وقتی زمان نمونه برداری را زیاد می‌کنیم باعث کند شدن پاسخ سیستم می‌شود. از طرفی کاهش بیش از حد زمان نمونه برداری باعث می‌شود که زمان نشست پاسخ افزایش یابد و از مقدار واقعی آن دور شویم.

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. 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.



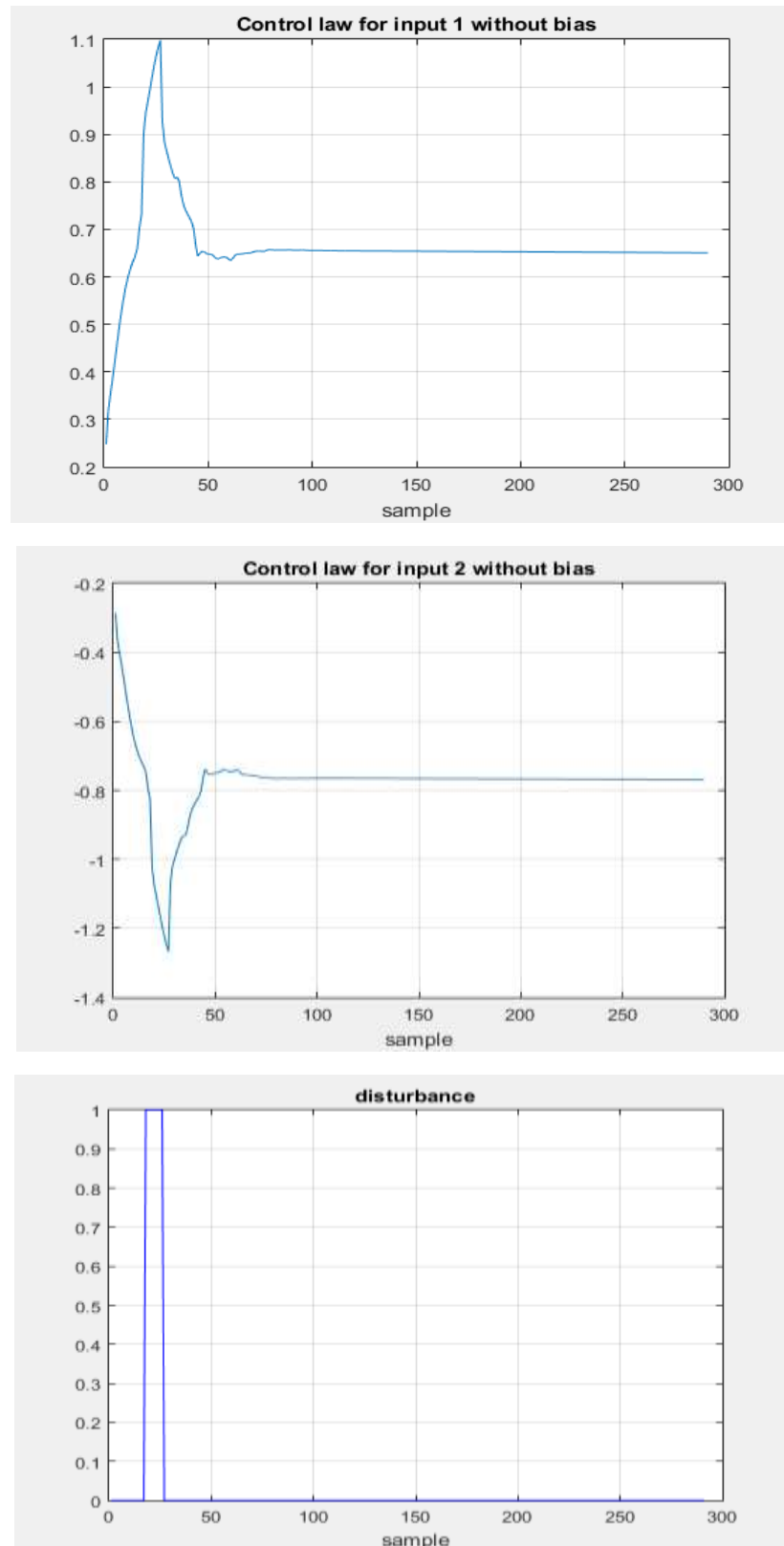
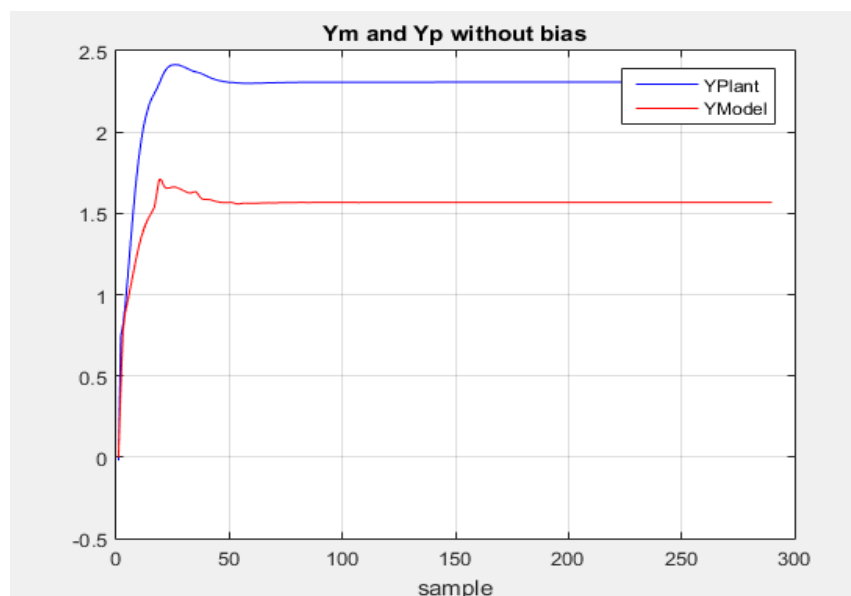
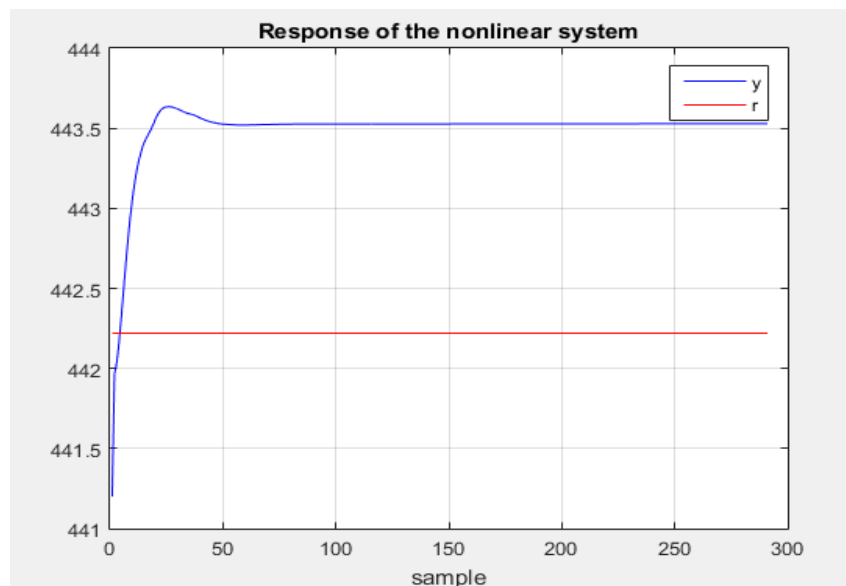


Figure 14. Responses of the system in verifying the effect of disturbance.

همانطور که در این شکل 14 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.

5.2. اثر نویز:

در این قسمت به خروجی پروسه نویز سفید با توان 0.1 اضافه می‌کنیم. پاسخ سیستم در حضور نویز سفید در زیر آمده است.



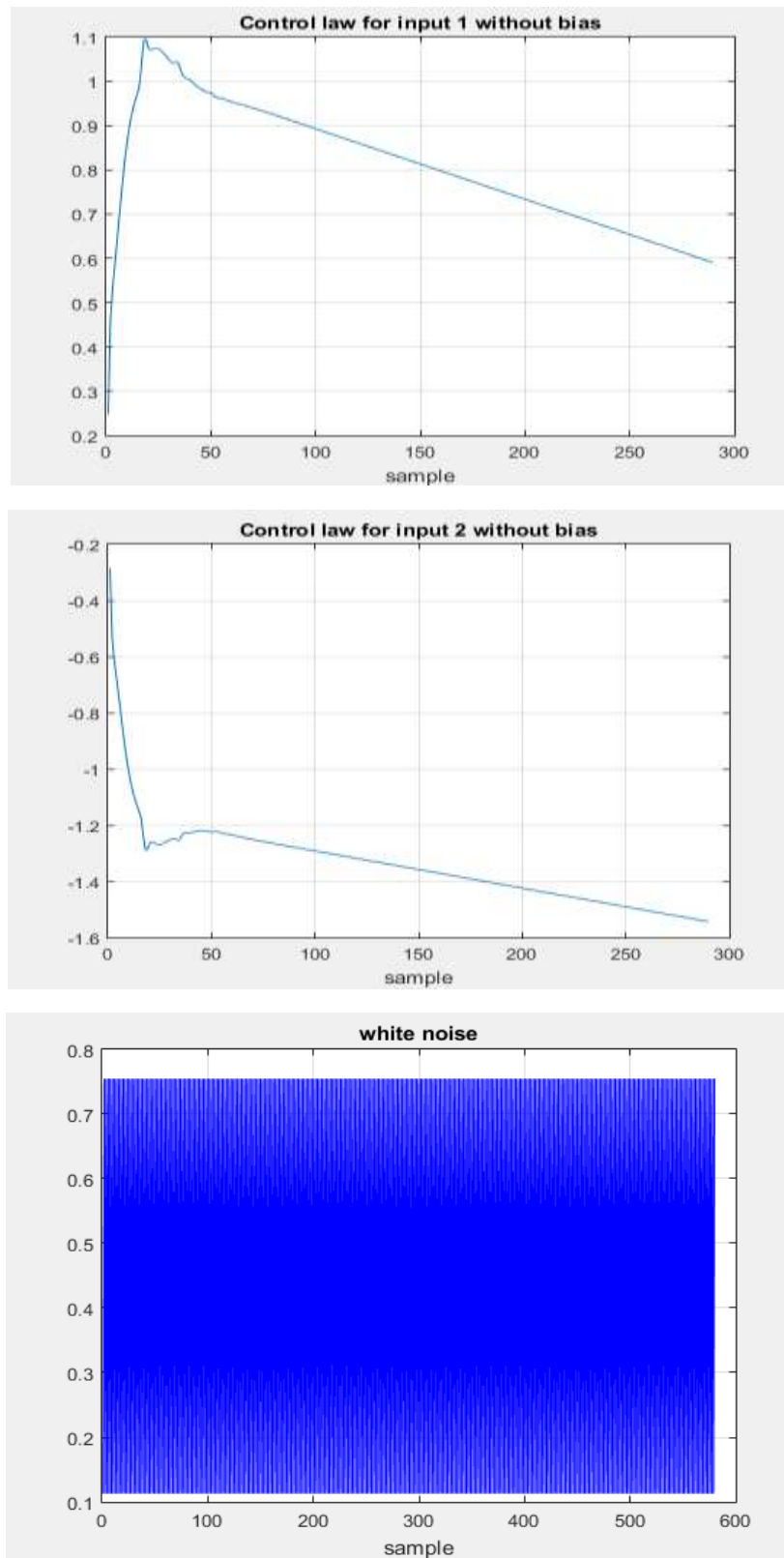
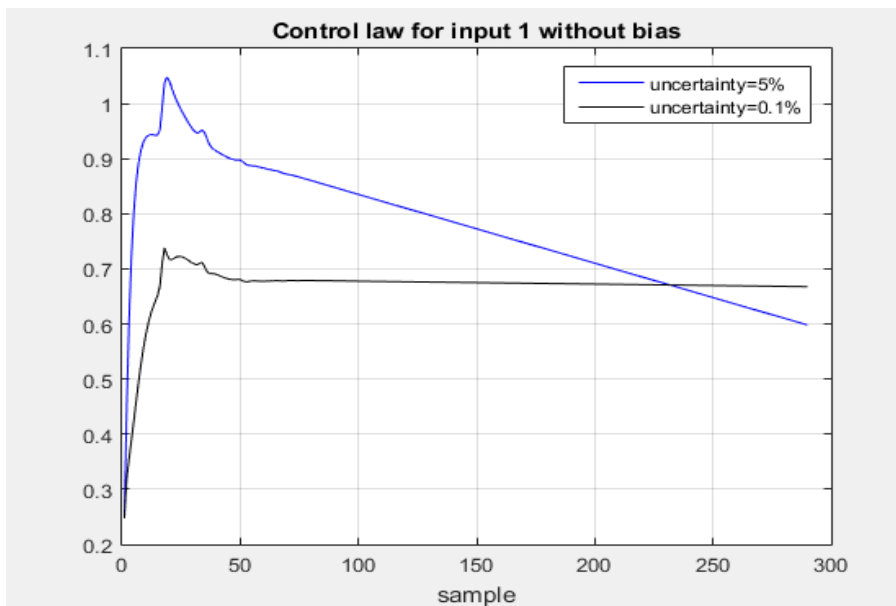
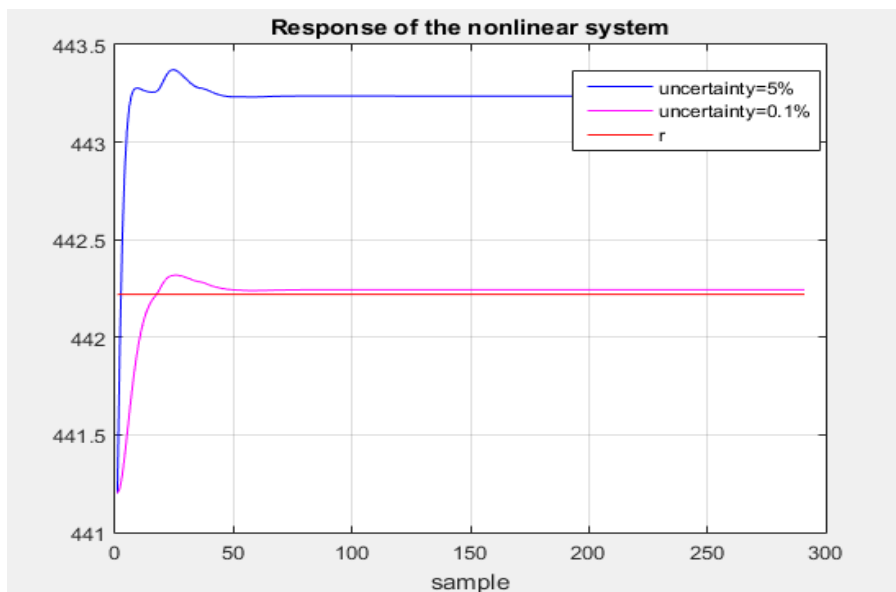


Figure 15. Responses of the system in verifying the effect of noise.

همانطور که در شکل بالا می بینیم نویز سفید باعث ایجاد بایاس در پاسخ سیستم شده است.

6. بررسی عدم قطعیت در پروسه:

در این بخش به بررسی عدم قطعیت بر روی پاسخ سیستم می پردازیم. ابتدا به پارامتر V سیستم غیر خطی که همان پروسه می باشد عدم قطعیت با درصد های مختلف اضافه می کنیم. برای این پارامتر عدم قطعیت با درصد های 5% و 0.1% اضافه می کنیم. پاسخ سیستم به ازای این عدم قطعیت ها برای این پارامتر در شکل زیر آورده شده است.



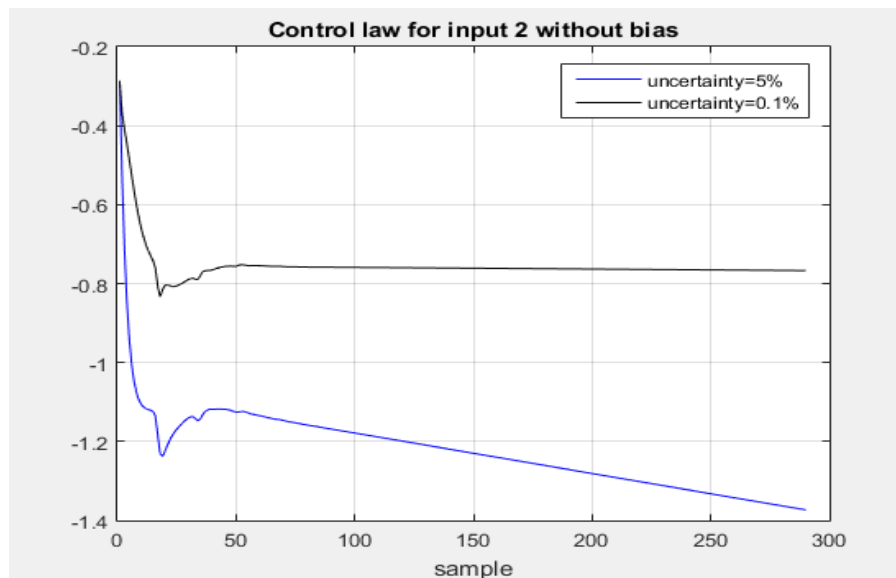
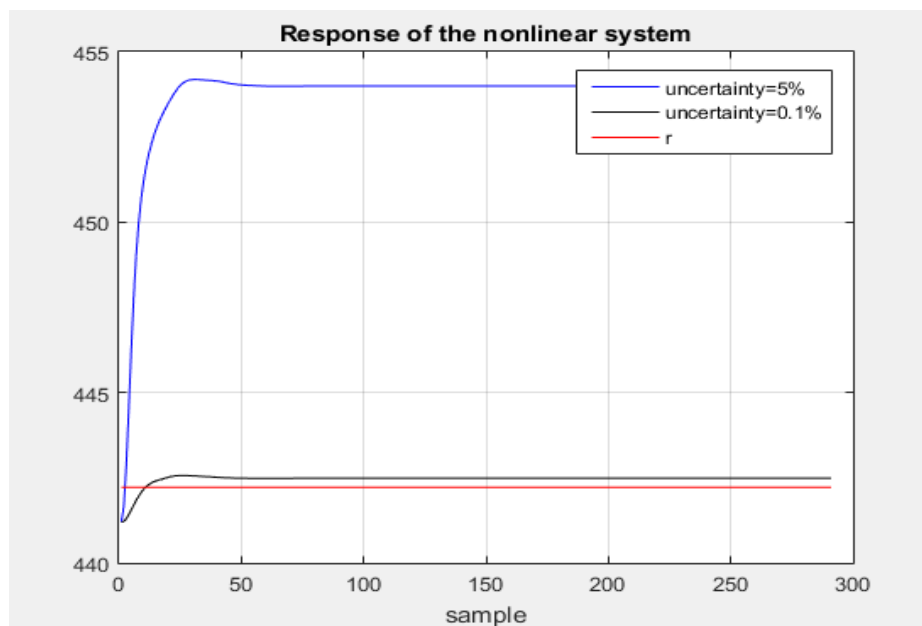


Figure 16. Responses of the system in verifying the effect of uncertainty on V .

همانطور که در شکل 16 می‌بینیم وجود عدم قطعیت باعث ایجاد خطای حالت ماندگار در پاسخ سیستم شده است. اما وقتی این درصد عدم قطعیت کم باشد خطایی ایجاد نمی‌کند یا این خطا بسیار کوچک خواهد بود.

اینبار عدم قطعیت با درصد های 5% و 0.1% را به پارامتر C_{A0} اضافه می‌کنیم. شکل زیر پاسخ سیستم در حضور عدم قطعیت های مذکور را نشان می‌دهد.



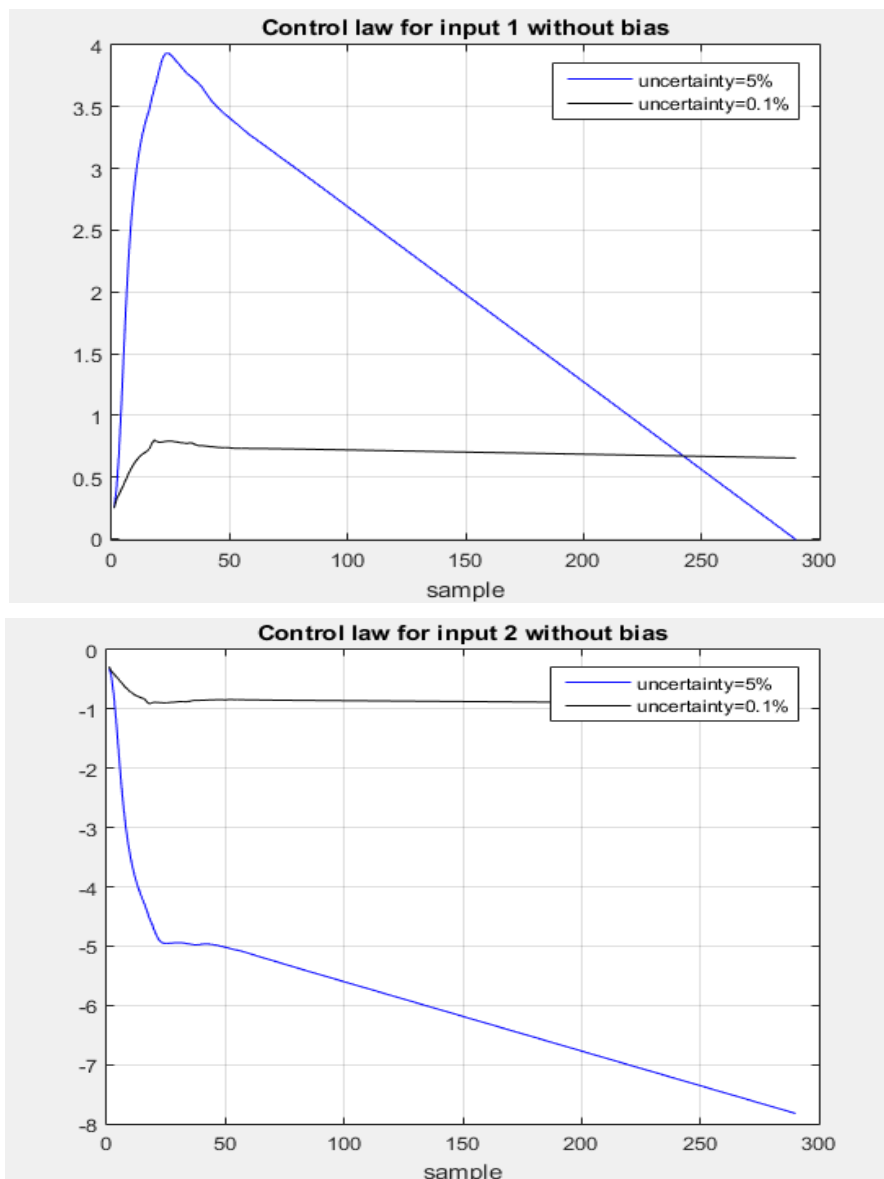


Figure 17. Responses of the system in verifying the effect of uncertainty on C_{A0} .

همانطور که در شکل 17 می‌بینیم عدم قطعیت روی پارامتر C_{A0} نیز مشابه V باعث ایجاد خطا در پاسخ سیستم در حالت ماندگار شد. اما همانطور که می‌بینیم خطایی که این پارامتر با وجود عدم قطعیت ایجاد کرده بیشتر از خطای عدم قطعیت روی V می‌باشد که این نشان دهنده این موضوع است که عدم قطعیت بستگی به حساسیت پارامتر تأثیرات متفاوتی روی پاسخ خواهد گذاشت.

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 parameter.....
% verify 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: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=P2;
N=N1;
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];
%.....Hankel Matrix.....
c1= [g1(3:P+2)];
r1 = [(g1(P+2:N+1))' zeros(1,P-1)];
G1_ = hankel(c1,r1);
%.....
c2 = [g2(3:P+2)];
r2 = [(g2(P+2:N+1))' zeros(1,P-1)];
G2_ = hankel(c2,r2);
G_=[G1_ G2_];
%.....
%%..... Designing.....
% alpha=0.1
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_DC^2*eye(M);
R=[R1 zeros(M); zeros(M) R2];
Kdmc=(G'*Q*G+R)\(G'*Q);
alpha=0.1;
x01=0.0882;
x02=441.2;
% .....
U1_ = zeros(P,length(t));
U2_ = zeros(P,length(t));
dU1_=zeros(N-1,length(t));
dU2_=zeros(N-1,length(t));
dU_=[dU1_;dU2_];
% U_=[U1_ ; U2_];
Ud1=zeros(N+P-1,length(t));
Ud2=zeros(N+P-1,length(t));
d=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));
U1=zeros(M,length(t));
U2=zeros(M,length(t));
% U=[U1;U2];
dU1=zeros(M,length(t));
dU2=zeros(M,length(t));
dU=[dU1;dU2];
% .....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)=G_*dU(:,i+1)+g1(N+1)*U1(:,i+1)+g2(N+1)*U2(:,i+1);
D(:,i+1)=d(i+1)*ones(P,1);

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

```



```

Ud1(2:N+P-1,i+2)=Ud1(1:N+P-2,i+1);
Ud1(1,i+2)=U1(1,i+1);
U1_(:,i+2)=Ud1(N:N+P-1,i+2);

```

```

Ud2(2:N+P-1,i+2)=Ud2(1:N+P-2,i+1);
Ud2(1,i+2)=U2(1,i+1);
U2_(:,i+2)=Ud2(N:N+P-1,i+2);

```

```

dU1_(2:N-1,i+2) = dU1_(1:N-2,i+1);
dU1_(1,i+2)=dU1(1,i+1);
dU2_(2:N-1,i+2) = dU2_(1:N-2,i+1);
dU2_(1,i+2)=dU2(1,i+1);

```

```

dU_(:,i+2)=[dU1_(:,i+2);dU2_(:,i+2)];

```

```

u1=U1(1,i+1);
u2=U2(1,i+1);
sim('Model')
% d(i+1)=y(end)-Y_m(1,i); % linear
d(i+1)=y(end)-Y_m(1,i);
% y=y(end); % linear
y=y(end); % +dist(i,1); % nonlinear
% y1=[y1;y(end)]; % linear
y1=[y1; y+441.2];
ym=[ym; Y_m(1,i)];
u_1=[u_1; u1];
u_2=[u_2; u2];
x01=x1(end);
x02=x2(end);
end
figure(1);
plot(y1,'b');
grid on
title('Response of the nonlinear system');
xlabel('sample');
figure(2);
plot(u_1,'b');
grid on
xlabel('sample');
title('Control law for input 1 without bias');
figure(3);
plot(u_2,'b');
grid on
xlabel('sample');
title('Control law for input 2 without bias');
% .....
% alpha=0.4
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];
Kdmc=(G'*Q*G+R)\(G'*Q);
alpha=0.4;
x01=0.0882;
x02=441.2;
% .....
U1_ = zeros(P,length(t));
U2_ = zeros(P,length(t));
dU1_=zeros(N-1,length(t));
dU2_=zeros(N-1,length(t));
dU_=[dU1_;dU2_];
% U_=[U1_ ; U2_];
Ud1=zeros(N+P-1,length(t));
Ud2=zeros(N+P-1,length(t));
d=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));
U1=zeros(M,length(t));
U2=zeros(M,length(t));
% U=[U1;U2];
dU1=zeros(M,length(t));
dU2=zeros(M,length(t));
dU=[dU1;dU2];
% .....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)=G_*dU(:,i+1)+g1(N+1)*U1(:,i+1)+g2(N+1)*U2(:,i+1);
D(:,i+1)=d(i+1)*ones(P,1);

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

Ud1(2:N+P-1,i+2)=Ud1(1:N+P-2,i+1);
Ud1(1,i+2)=U1(1,i+1);

```

```

U1_(:,i+2)=Ud1(N:N+P-1,i+2);

Ud2(2:N+P-1,i+2)=Ud2(1:N+P-2,i+1);
Ud2(1,i+2)=U2(1,i+1);
U2_(:,i+2)=Ud2(N:N+P-1,i+2);

dU1_(2:N-1,i+2) = dU1_(1:N-2,i+1);
dU1_(1,i+2)=dU1(1,i+1);
dU2_(2:N-1,i+2) = dU2_(1:N-2,i+1);
dU2_(1,i+2)=dU2(1,i+1);

dU_(:,i+2)=[dU1_(:,i+2);dU2_(:,i+2)];

u1=U1(1,i+1);
u2=U2(1,i+1);
sim('Model')
%d(i+1)=yl(end)-Y_m(1,i); %linear
d(i+1)=y(end)-Y_m(1,i);
%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];
x01=x1(end);
x02=x2(end);
end
figure(1);
hold on
plot(y1,'m');
figure(2);
hold on
plot(u_1,'m');
figure(3);
hold on
plot(u_2,'m');
% .....
%alpha=0.6
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];
Kdmc=(G'*Q*G+R)\(G'*Q);
alpha=0.6;
x01=0.0882;
x02=441.2;
% .....
U1_ = zeros(P,length(t));
U2_ = zeros(P,length(t));

```

```

dU1_=zeros(N-1,length(t));
dU2_=zeros(N-1,length(t));
dU_=[dU1_;dU2_];
% U_=[U1_ ; U2_];
Ud1=zeros(N+P-1,length(t));
Ud2=zeros(N+P-1,length(t));
d=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));
U1=zeros(M,length(t));
U2=zeros(M,length(t));
% U=[U1;U2];
dU1=zeros(M,length(t));
dU2=zeros(M,length(t));
dU=[dU1;dU2];
% .....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)=G_*dU(:,i+1)+g1(N+1)*U1(:,i+1)+g2(N+1)*U2(:,i+1);
D(:,i+1)=d(i+1)*ones(P,1);

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

Ud1(2:N+P-1,i+2)=Ud1(1:N+P-2,i+1);
Ud1(1,i+2)=U1(1,i+1);
U1_(:,i+2)=Ud1(N:N+P-1,i+2);

Ud2(2:N+P-1,i+2)=Ud2(1:N+P-2,i+1);
Ud2(1,i+2)=U2(1,i+1);
U2_(:,i+2)=Ud2(N:N+P-1,i+2);

```

```

dU1_(2:N-1,i+2) = dU1_(1:N-2,i+1);
dU1_(1,i+2)=dU1(1,i+1);
dU2_(2:N-1,i+2) = dU2_(1:N-2,i+1);
dU2_(1,i+2)=dU2(1,i+1);

dU_(:,i+2)=[dU1_(:,i+2);dU2_(:,i+2)];

u1=U1(1,i+1);
u2=U2(1,i+1);
sim('Model')
%d(i+1)=yl(end)-Y_m(1,i); %linear
d(i+1)=y(end)-Y_m(1,i);
%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];
x01=x1(end);
x02=x2(end);
end
figure(1);
hold on
plot(y1,'c');
hold on
plot(r+441.22,'r')
legend('alpha=0.1','alpha=0.4','alpha=0.6','r');
figure(2);
hold on
plot(u_1,'c');
legend('alpha=0.1','alpha=0.4','alpha=0.6');
figure(3);
hold on
plot(u_2,'c');
legend('alpha=0.1','alpha=0.4','alpha=0.6');
% .....
%% .....
% .....verifying parameter.....
% verify gamma
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: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=P2;
N=N1;
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];
% .....Hankel Matrix.....
c1= [g1(3:P+2)];
r1 = [(g1(P+2:N+1))' zeros(1,P-1)];
G1_ = hankel(c1,r1);
% .....
c2 = [g2(3:P+2)];
r2 = [(g2(P+2:N+1))' zeros(1,P-1)];
G2_ = hankel(c2,r2);
G_=[G1_ G2_];
% .....
%% ..... Designing.....
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];
Kdmc=(G'*Q*(G+R))/(G'*Q);
alpha=0.5;
x01=0.0882;
x02=441.2;
% .....
U1_ = zeros(P,length(t));
U2_ = zeros(P,length(t));
dU1_=zeros(N-1,length(t));
dU2_=zeros(N-1,length(t));
dU_=[dU1_;dU2_];
% U_=[U1_ ; U2_];
Ud1=zeros(N+P-1,length(t));

```

```

Ud2=zeros(N+P-1,length(t));
d=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));
U1=zeros(M,length(t));
U2=zeros(M,length(t));
% U=[U1;U2];
dU1=zeros(M,length(t));
dU2=zeros(M,length(t));
dU=[dU1;dU2];
%.....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)=G_*dU(:,i+1)+g1(N+1)*U1(:,i+1)+g2(N+1)*U2(:,i+1);
D(:,i+1)=d(i+1)*ones(P,1);

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

Ud1(2:N+P-1,i+2)=Ud1(1:N+P-2,i+1);
Ud1(1,i+2)=U1(1,i+1);
U1_(:,i+2)=Ud1(N:N+P-1,i+2);

Ud2(2:N+P-1,i+2)=Ud2(1:N+P-2,i+1);
Ud2(1,i+2)=U2(1,i+1);
U2_(:,i+2)=Ud2(N:N+P-1,i+2);

dU1_(2:N-1,i+2) = dU1_(1:N-2,i+1);
dU1_(1,i+2)=dU1(1,i+1);
dU2_(2:N-1,i+2) = dU2_(1:N-2,i+1);
dU2_(1,i+2)=dU2(1,i+1);

```

```

dU_(:,i+2)=[dU1_(:,i+2);dU2_(:,i+2)];

u1=U1(1,i+1);
u2=U2(1,i+1);
sim('Model')
% d(i+1)=y1(end)-Y_m(1,i); % linear
d(i+1)=y(end)-Y_m(1,i);
% y=y1(end); % linear
y=y(end); % +dist(i,1); % nonlinear
% y1=[y1;y1(end)]; % linear
y1=[y1; y+441.2];
ym=[ym; Y_m(1,i)];
u_1=[u_1; u1];
u_2=[u_2; u2];
x01=x1(end);
x02=x2(end);
end
figure(1);
plot(y1,'b');
grid on
title('Response of the nonlinear system');
xlabel('sample');
figure(2);
plot(u_1,'b');
grid on
xlabel('sample');
title('Control law for input 1 without bias');
figure(3);
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];
Kdmc=(G'*Q*(G+R)\(G'*Q));
alpha=0.5;
x01=0.0882;
x02=441.2;
% .....
U1_ = zeros(P,length(t));
U2_ = zeros(P,length(t));
dU1_=zeros(N-1,length(t));
dU2_=zeros(N-1,length(t));
dU_=[dU1_;dU2_];
% U_=[U1_ ; U2_];
Ud1=zeros(N+P-1,length(t));
Ud2=zeros(N+P-1,length(t));

```



```

d=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));
U1=zeros(M,length(t));
U2=zeros(M,length(t));
% U=[U1;U2];
dU1=zeros(M,length(t));
dU2=zeros(M,length(t));
dU=[dU1;dU2];
%.....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)=G_*dU(:,i+1)+g1(N+1)*U1(:,i+1)+g2(N+1)*U2(:,i+1);
D(:,i+1)=d(i+1)*ones(P,1);

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

Ud1(2:N+P-1,i+2)=Ud1(1:N+P-2,i+1);
Ud1(1,i+2)=U1(1,i+1);
U1_(:,i+2)=Ud1(N:N+P-1,i+2);

Ud2(2:N+P-1,i+2)=Ud2(1:N+P-2,i+1);
Ud2(1,i+2)=U2(1,i+1);
U2_(:,i+2)=Ud2(N:N+P-1,i+2);

dU1_(2:N-1,i+2) = dU1_(1:N-2,i+1);
dU1_(1,i+2)=dU1(1,i+1);
dU2_(2:N-1,i+2) = dU2_(1:N-2,i+1);
dU2_(1,i+2)=dU2(1,i+1);

```

```

dU_(:,i+2)=[dU1_(:,i+2);dU2_(:,i+2)];

u1=U1(1,i+1);
u2=U2(1,i+1);
sim('Model')
%d(i+1)=yl(end)-Y_m(1,i); %linear
d(i+1)=y(end)-Y_m(1,i);
%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];
x01=x1(end);
x02=x2(end);
end
figure(1);
hold on
plot(y1,'m');
figure(2);
hold on
plot(u_1,'m');
figure(3);
hold on
plot(u_2,'m');
%.....
% 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];
Kdmc=(G'*Q*G+R)\(G'*Q);
alpha=0.5;
x01=0.0882;
x02=441.2;
%.....
U1_ = zeros(P,length(t));
U2_ = zeros(P,length(t));
dU1_=zeros(N-1,length(t));
dU2_=zeros(N-1,length(t));
dU_=[dU1_;dU2_];
% U_=[U1_ ; U2_];
Ud1=zeros(N+P-1,length(t));
Ud2=zeros(N+P-1,length(t));
d=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));
U1=zeros(M,length(t));
U2=zeros(M,length(t));
% U=[U1;U2];
dU1=zeros(M,length(t));
dU2=zeros(M,length(t));
dU=[dU1;dU2];
% .....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)=G_*dU(:,i+1)+g1(N+1)*U1(:,i+1)+g2(N+1)*U2(:,i+1);
D(:,i+1)=d(i+1)*ones(P,1);

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

Ud1(2:N+P-1,i+2)=Ud1(1:N+P-2,i+1);
Ud1(1,i+2)=U1(1,i+1);
U1_(:,i+2)=Ud1(N:N+P-1,i+2);

Ud2(2:N+P-1,i+2)=Ud2(1:N+P-2,i+1);
Ud2(1,i+2)=U2(1,i+1);
U2_(:,i+2)=Ud2(N:N+P-1,i+2);

dU1_(2:N-1,i+2) = dU1_(1:N-2,i+1);
dU1_(1,i+2)=dU1(1,i+1);
dU2_(2:N-1,i+2) = dU2_(1:N-2,i+1);
dU2_(1,i+2)=dU2(1,i+1);

dU_(:,i+2)=[dU1_(:,i+2);dU2_(:,i+2)];

u1=U1(1,i+1);
u2=U2(1,i+1);
sim('Model')
%d(i+1)=yl(end)-Y_m(1,i); % linear
d(i+1)=y(end)-Y_m(1,i);
%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];
x01=x1(end);
x02=x2(end);
end
figure(1);
hold on
plot(y1,'c');
hold on
plot(r+441.22,'r')
legend('gamma=1','gamma=1/2','gamma=1/600','r');
figure(2);
hold on
plot(u_1,'c');
legend('gamma=1','gamma=1/2','gamma=1/600');
figure(3);
hold on
plot(u_2,'c');
legend('gamma=1','gamma=1/2','gamma=1/600');
% .....
%% .....
% .....verifying parameter.....
% verify N
% N=40
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: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=P2;
N=40;
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];
%.....Hankel Matrix.....
c1= [g1(3:P+2)];
r1 = [(g1(P+2:N+1))' zeros(1,P-1)];
G1_ = hankel(c1,r1);
%.....
c2 = [g2(3:P+2)];
r2 = [(g2(P+2:N+1))' zeros(1,P-1)];
G2_ = hankel(c2,r2);
G_=[G1_ G2_];
%.....
%%..... Designing.....
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];
Kdmc=(G'*Q*G+R)\(G'*Q);
alpha=0.5;
x01=0.0882;
x02=441.2;
%.....
U1_ = zeros(P,length(t));
U2_ = zeros(P,length(t));
dU1_=zeros(N-1,length(t));
dU2_=zeros(N-1,length(t));
dU_=[dU1_;dU2_];
% U_=[U1_ ; U2_];
Ud1=zeros(N+P-1,length(t));
Ud2=zeros(N+P-1,length(t));
d=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));
U1=zeros(M,length(t));

```

```

U2=zeros(M,length(t));
% U=[U1;U2];
dU1=zeros(M,length(t));
dU2=zeros(M,length(t));
dU=[dU1;dU2];
% .....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)=G_*dU(:,i+1)+g1(N+1)*U1(:,i+1)+g2(N+1)*U2(:,i+1);
D(:,i+1)=d(i+1)*ones(P,1);

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

Ud1(2:N+P-1,i+2)=Ud1(1:N+P-2,i+1);
Ud1(1,i+2)=U1(1,i+1);
U1_(:,i+2)=Ud1(N:N+P-1,i+2);

Ud2(2:N+P-1,i+2)=Ud2(1:N+P-2,i+1);
Ud2(1,i+2)=U2(1,i+1);
U2_(:,i+2)=Ud2(N:N+P-1,i+2);

dU1_(2:N-1,i+2) = dU1_(1:N-2,i+1);
dU1_(1,i+2)=dU1(1,i+1);
dU2_(2:N-1,i+2) = dU2_(1:N-2,i+1);
dU2_(1,i+2)=dU2(1,i+1);

dU_(:,i+2)=[dU1_(:,i+2);dU2_(:,i+2)];

u1=U1(1,i+1);
u2=U2(1,i+1);
sim('Model')
%d(i+1)=yl(end)-Y_m(1,i); %linear
d(i+1)=y(end)-Y_m(1,i);
%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];
x01=x1(end);
x02=x2(end);
end
figure(1);
plot(y1,'b');
grid on
title('Response of the nonlinear system');
xlabel('sample');
figure(2);
plot(u_1,'b');
grid on
xlabel('sample');
title('Control law for input 1 without bias');
figure(3);
plot(u_2,'b');
grid on
xlabel('sample');
title('Control law for input 2 without bias');
% .....
% N=17
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: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=P2;
N=N1;
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];
%.....Hankel Matrix.....
c1= [g1(3:P+2)];
r1 = [(g1(P+2:N+1))' zeros(1,P-1)];
G1_ = hankel(c1,r1);
%.....
c2 = [g2(3:P+2)];
r2 = [(g2(P+2:N+1))' zeros(1,P-1)];
G2_ = hankel(c2,r2);
G_=[G1_ G2_];
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];
Kdmc=(G'*Q*G+R)\(G'*Q);
alpha=0.5;
x01=0.0882;
x02=441.2;
%.....
U1_ = zeros(P,length(t));
U2_ = zeros(P,length(t));
dU1_ = zeros(N-1,length(t));
dU2_ = zeros(N-1,length(t));
dU_=[dU1_;dU2_];
% U_=[U1_ ; U2_];
Ud1=zeros(N+P-1,length(t));
Ud2=zeros(N+P-1,length(t));
d=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));
U1=zeros(M,length(t));
U2=zeros(M,length(t));
% U=[U1;U2];
dU1=zeros(M,length(t));
dU2=zeros(M,length(t));
dU=[dU1;dU2];
%.....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)=G_*dU(:,i+1)+g1(N+1)*U1(:,i+1)+g2(N+1)*U2(:,i+1);
D(:,i+1)=d(i+1)*ones(P,1);

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

Ud1(2:N+P-1,i+2)=Ud1(1:N+P-2,i+1);
Ud1(1,i+2)=U1(1,i+1);
U1_(:,i+2)=Ud1(N:N+P-1,i+2);

Ud2(2:N+P-1,i+2)=Ud2(1:N+P-2,i+1);
Ud2(1,i+2)=U2(1,i+1);
U2_(:,i+2)=Ud2(N:N+P-1,i+2);

dU1_(2:N-1,i+2) = dU1_(1:N-2,i+1);
dU1_(1,i+2)=dU1(1,i+1);
dU2_(2:N-1,i+2) = dU2_(1:N-2,i+1);
dU2_(1,i+2)=dU2(1,i+1);

dU_(:,i+2)=[dU1_(:,i+2);dU2_(:,i+2)];

u1=U1(1,i+1);
u2=U2(1,i+1);
sim('Model')
%d(i+1)=yl(end)-Y_m(1,i); %linear
d(i+1)=y(end)-Y_m(1,i);
%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];
x01=x1(end);
x02=x2(end);
end
figure(1);
hold on
plot(y1,'m');
figure(2);
hold on

```

```

plot(u_1,'m');
figure(3);
hold on
plot(u_2,'m');
% .....
% N=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: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=P2;
N=5;
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];
% .....Hankel Matrix.....
c1= [g1(3:P+2)];
r1 = [(g1(P+2:N+1))' zeros(1,P-1)];
G1_ = hankel(c1,r1);
% .....
c2 = [g2(3:P+2)];
r2 = [(g2(P+2:N+1))' zeros(1,P-1)];
G2_ = hankel(c2,r2);
G_=[G1_ G2_];
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];
Kdmc=(G'*Q*G+R)\(G'*Q);
alpha=0.5;
x01=0.0882;
x02=441.2;
% .....
U1_ = zeros(P,length(t));
U2_ = zeros(P,length(t));
dU1_=zeros(N-1,length(t));
dU2_=zeros(N-1,length(t));
dU_=[dU1_;dU2_];
% U_=[U1_ ; U2_];
Ud1=zeros(N+P-1,length(t));
Ud2=zeros(N+P-1,length(t));
d=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));
U1=zeros(M,length(t));
U2=zeros(M,length(t));
% U=[U1;U2];
dU1=zeros(M,length(t));
dU2=zeros(M,length(t));
dU=[dU1;dU2];
% .....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)=G_*dU(:,i+1)+g1(N+1)*U1(:,i+1)+g2(N+1)*U2(:,i+1);
D(:,i+1)=d(i+1)*ones(P,1);

E(:,i+1)=Y_d(:,i+1)-Y_past(:,i+1)-D(:,i+1);
dU(:,i+1)=Kdmc*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);
```

```
Ud1(2:N+P-1,i+2)=Ud1(1:N+P-2,i+1);
Ud1(1,i+2)=U1(1,i+1);
U1_(:,i+2)=Ud1(N:N+P-1,i+2);
```

```
Ud2(2:N+P-1,i+2)=Ud2(1:N+P-2,i+1);
Ud2(1,i+2)=U2(1,i+1);
U2_(:,i+2)=Ud2(N:N+P-1,i+2);
```

```
dU1_(2:N-1,i+2) = dU1_(1:N-2,i+1);
dU1_(1,i+2)=dU1(1,i+1);
dU2_(2:N-1,i+2) = dU2_(1:N-2,i+1);
dU2_(1,i+2)=dU2(1,i+1);
```

```
dU_(:,i+2)=[dU1_(:,i+2);dU2_(:,i+2)];
```

```
u1=U1(1,i+1);
u2=U2(1,i+1);
sim('Model')
% d(i+1)=y(end)-Y_m(1,i); % linear
d(i+1)=y(end)-Y_m(1,i);
% y=y(end); % linear
y=y(end); % +dist(i,1); % nonlinear
% y1=[y1;y(end)]; % linear
y1=[y1; y+441.2];
ym=[ym; Y_m(1,i)];
u_1=[u_1; u1];
u_2=[u_2; u2];
x01=x1(end);
x02=x2(end);
end
figure(1);
hold on
plot(y1,'c');
hold on
plot(r+441.22,'r')
legend('N=40','N=17','N=5','r');
figure(2);
hold on
plot(u_1,'c');
legend('N=40','N=17','N=5');
figure(3);
hold on
plot(u_2,'c');
legend('N=40','N=17','N=5');
% .....
% % .....
% .....verifying parameter.....
% verify P or M
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: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=P2;
N=N1;
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];
%.....Hankel Matrix.....
c1= [g1(3:P+2)];
r1 = [(g1(P+2:N+1))' zeros(1,P-1)];
G1_ = hankel(c1,r1);
%.....
c2 = [g2(3:P+2)];
r2 = [(g2(P+2:N+1))' zeros(1,P-1)];
G2_ = hankel(c2,r2);
G_=[G1_ G2_];
%.....
%%..... Designing.....
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];
Kdmc=(G'*Q*G+R)\(G'*Q);
alpha=0.5;
x01=0.0882;
x02=441.2;
% .....
U1_ = zeros(P,length(t));
U2_ = zeros(P,length(t));
dU1_=zeros(N-1,length(t));
dU2_=zeros(N-1,length(t));
dU_=[dU1_;dU2_];
% U_=[U1_ ; U2_];
Ud1=zeros(N+P-1,length(t));
Ud2=zeros(N+P-1,length(t));
d=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));
U1=zeros(M,length(t));
U2=zeros(M,length(t));
% U=[U1;U2];
dU1=zeros(M,length(t));
dU2=zeros(M,length(t));
dU=[dU1;dU2];
% .....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)=G_*dU(:,i+1)+g1(N+1)*U1(:,i+1)+g2(N+1)*U2(:,i+1);
D(:,i+1)=d(i+1)*ones(P,1);

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

Ud1(2:N+P-1,i+2)=Ud1(1:N+P-2,i+1);
Ud1(1,i+2)=U1(1,i+1);

```

```

U1_(:,i+2)=Ud1(N:N+P-1,i+2);

Ud2(2:N+P-1,i+2)=Ud2(1:N+P-2,i+1);
Ud2(1,i+2)=U2(1,i+1);
U2_(:,i+2)=Ud2(N:N+P-1,i+2);


dU1_(2:N-1,i+2) = dU1_(1:N-2,i+1);
dU1_(1,i+2)=dU1(1,i+1);
dU2_(2:N-1,i+2) = dU2_(1:N-2,i+1);
dU2_(1,i+2)=dU2(1,i+1);


dU_(:,i+2)=[dU1_(:,i+2);dU2_(:,i+2)];


u1=U1(1,i+1);
u2=U2(1,i+1);
sim('Model')
% d(i+1)=yl(end)-Y_m(1,i); %linear
d(i+1)=y(end)-Y_m(1,i);
% 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];
x01=x1(end);
x02=x2(end);
end
figure(1);
plot(y1,'b');
grid on
title('Response of the nonlinear system');
xlabel('sample');
figure(2);
plot(u_1,'b');
grid on
xlabel('sample');
title('Control law for input 1 without bias');
figure(3);
plot(u_2,'b');
grid on
xlabel('sample');
title('Control law for input 2 without bias');
% .....
% 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: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=2;
N=N1;
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];
% .....Hankel Matrix.....
c1= [g1(3:P+2)];
r1 = [(g1(P+2:N+1))' zeros(1,P-1)];
G1_ = hankel(c1,r1);
% .....
c2 = [g2(3:P+2)];
r2 = [(g2(P+2:N+1))' zeros(1,P-1)];
G2_ = hankel(c2,r2);
G_=[G1_ G2_];
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];
Kdmc=(G'*Q*G+R)\(G'*Q);
alpha=0.5;
x01=0.0882;
x02=441.2;
% .....
U1_ = zeros(P,length(t));
U2_ = zeros(P,length(t));
dU1_=zeros(N-1,length(t));

```



```

dU2_=zeros(N-1,length(t));
dU_=[dU1_;dU2_];
% U_=[U1_ ; U2_];
Ud1=zeros(N+P-1,length(t));
Ud2=zeros(N+P-1,length(t));
d=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));
U1=zeros(M,length(t));
U2=zeros(M,length(t));
% U=[U1;U2];
dU1=zeros(M,length(t));
dU2=zeros(M,length(t));
dU=[dU1;dU2];
% .....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)=G_*dU(:,i+1)+g1(N+1)*U1_(:,i+1)+g2(N+1)*U2_(:,i+1);
D(:,i+1)=d(i+1)*ones(P,1);

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

Ud1(2:N+P-1,i+2)=Ud1(1:N+P-2,i+1);
Ud1(1,i+2)=U1(1,i+1);
U1_(:,i+2)=Ud1(N:N+P-1,i+2);

Ud2(2:N+P-1,i+2)=Ud2(1:N+P-2,i+1);
Ud2(1,i+2)=U2(1,i+1);
U2_(:,i+2)=Ud2(N:N+P-1,i+2);

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

```

```

dU1_(1,i+2)=dU1(1,i+1);
dU2_(2:N-1,i+2) = dU2_(1:N-2,i+1);
dU2_(1,i+2)=dU2(1,i+1);

dU_(:,i+2)=[dU1_(:,i+2);dU2_(:,i+2)];

u1=U1(1,i+1);
u2=U2(1,i+1);
sim('Model')
%d(i+1)=yl(end)-Y_m(1,i); %linear
d(i+1)=y(end)-Y_m(1,i);
%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];
x01=x1(end);
x02=x2(end);
end
figure(1);
hold on
plot(y1,'m');
figure(2);
hold on
plot(u_1,'m');
figure(3);
hold on
plot(u_2,'m');
%.....
% 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: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=15;
N=N1;
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];
%.....Hankel Matrix.....
c1= [g1(3:P+2)];
r1 = [(g1(P+2:N+1))' zeros(1,P-1)];
G1_ = hankel(c1,r1);
%.....
c2 = [g2(3:P+2)];
r2 = [(g2(P+2:N+1))' zeros(1,P-1)];
G2_ = hankel(c2,r2);
G_=[G1_ G2_];
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];
Kdmc=(G'*Q*G+R)\(G'*Q);
alpha=0.5;
x01=0.0882;
x02=441.2;
%.....
U1_ = zeros(P,length(t));
U2_ = zeros(P,length(t));
dU1_=zeros(N-1,length(t));
dU2_=zeros(N-1,length(t));
dU_=[dU1_;dU2_];
% U_=[U1_ ; U2_];
Ud1=zeros(N+P-1,length(t));
Ud2=zeros(N+P-1,length(t));
d=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));
U1=zeros(M,length(t));
U2=zeros(M,length(t));
% U=[U1;U2];
dU1=zeros(M,length(t));
dU2=zeros(M,length(t));
dU=[dU1;dU2];
%.....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)=G_*dU(:,i+1)+g1(N+1)*U1(:,i+1)+g2(N+1)*U2(:,i+1);
D(:,i+1)=d(i+1)*ones(P,1);

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

Ud1(2:N+P-1,i+2)=Ud1(1:N+P-2,i+1);
Ud1(1,i+2)=U1(1,i+1);
U1(:,i+2)=Ud1(N:N+P-1,i+2);

Ud2(2:N+P-1,i+2)=Ud2(1:N+P-2,i+1);
Ud2(1,i+2)=U2(1,i+1);
U2(:,i+2)=Ud2(N:N+P-1,i+2);

dU1_(2:N-1,i+2) = dU1_(1:N-2,i+1);
dU1_(1,i+2)=dU1(1,i+1);
dU2_(2:N-1,i+2) = dU2_(1:N-2,i+1);
dU2_(1,i+2)=dU2(1,i+1);

dU_(:,i+2)=[dU1_(:,i+2);dU2_(:,i+2)];

u1=U1(1,i+1);
u2=U2(1,i+1);
sim('Model')
%d(i+1)=yl(end)-Y_m(1,i); % linear
d(i+1)=y(end)-Y_m(1,i);
%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];
x01=x1(end);
x02=x2(end);
end
figure(1);
hold on
plot(y1,'c');
hold on
plot(r+441.22,'r')
legend('P=5','P=2','P=15','r');
figure(2);
hold on
plot(u_1,'c');
legend('P=5','P=2','P=15');
figure(3);
hold on
plot(u_2,'c');
legend('P=5','P=2','P=15');
% .....
% % .....
% .....verifying parameter.....
% verify 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: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=P2;
N=N1;
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];
%.....Hankel Matrix.....
c1= [g1(3:P+2)];
r1 = [(g1(P+2:N+1))' zeros(1,P-1)];
G1_ = hankel(c1,r1);
%.....
c2 = [g2(3:P+2)];
r2 = [(g2(P+2:N+1))' zeros(1,P-1)];
G2_ = hankel(c2,r2);
G_=[G1_ G2_];
%.....
%%..... Designing.....
% 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];
Kdmc=(G'*Q*G+R)\(G'*Q);
alpha=0.5;
x01=0.0882;
x02=441.2;
%.....
U1_ = zeros(P,length(t));
U2_ = zeros(P,length(t));
dU1_=zeros(N-1,length(t));
dU2_=zeros(N-1,length(t));
dU_=[dU1_;dU2_];
% U_=[U1_ ; U2_];
Ud1=zeros(N+P-1,length(t));
Ud2=zeros(N+P-1,length(t));
d=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));
U1=zeros(M,length(t));
U2=zeros(M,length(t));

```

```

% U=[U1;U2];
dU1=zeros(M,length(t));
dU2=zeros(M,length(t));
dU=[dU1;dU2];
% .....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)=G_*dU(:,i+1)+g1(N+1)*U1(:,i+1)+g2(N+1)*U2(:,i+1);
D(:,i+1)=d(i+1)*ones(P,1);

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

Ud1(2:N+P-1,i+2)=Ud1(1:N+P-2,i+1);
Ud1(1,i+2)=U1(1,i+1);
U1_(:,i+2)=Ud1(N:N+P-1,i+2);

Ud2(2:N+P-1,i+2)=Ud2(1:N+P-2,i+1);
Ud2(1,i+2)=U2(1,i+1);
U2_(:,i+2)=Ud2(N:N+P-1,i+2);

dU1_(2:N-1,i+2) = dU1_(1:N-2,i+1);
dU1_(1,i+2)=dU1(1,i+1);
dU2_(2:N-1,i+2) = dU2_(1:N-2,i+1);
dU2_(1,i+2)=dU2(1,i+1);

dU_(:,i+2)=[dU1_(:,i+2);dU2_(:,i+2)];

u1=U1(1,i+1);
u2=U2(1,i+1);
sim('Model')
%d(i+1)=yl(end)-Y_m(1,i); %linear
d(i+1)=y(end)-Y_m(1,i);
%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];

```

```

x01=x1(end);
x02=x2(end);
end
figure(1);
plot(y1,'b');
grid on
title('Response of the nonlinear system');
xlabel('sample');
figure(2);
plot(u_1,'b');
grid on
xlabel('sample');
title('Control law for input 1 without bias');
figure(3);
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];
Kdmc=(G'*Q*G+R)\(G'*Q);
alpha=0.5;
x01=0.0882;
x02=441.2;
% .....
U1_ = zeros(P,length(t));
U2_ = zeros(P,length(t));
dU1_=zeros(N-1,length(t));
dU2_=zeros(N-1,length(t));
dU_=[dU1_;dU2_];
% U_=[U1_ ; U2_];
Ud1=zeros(N+P-1,length(t));
Ud2=zeros(N+P-1,length(t));
d=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));
U1=zeros(M,length(t));
U2=zeros(M,length(t));
% U=[U1;U2];
dU1=zeros(M,length(t));

```



```

dU2=zeros(M,length(t));
dU=[dU1;dU2];
%.....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)=G_*dU(:,i+1)+g1(N+1)*U1_(:,i+1)+g2(N+1)*U2_(:,i+1);
D(:,i+1)=d(i+1)*ones(P,1);

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

Ud1(2:N+P-1,i+2)=Ud1(1:N+P-2,i+1);
Ud1(1,i+2)=U1(1,i+1);
U1_(:,i+2)=Ud1(N:N+P-1,i+2);

Ud2(2:N+P-1,i+2)=Ud2(1:N+P-2,i+1);
Ud2(1,i+2)=U2(1,i+1);
U2_(:,i+2)=Ud2(N:N+P-1,i+2);

dU1_(2:N-1,i+2) = dU1_(1:N-2,i+1);
dU1_(1,i+2)=dU1(1,i+1);
dU2_(2:N-1,i+2) = dU2_(1:N-2,i+1);
dU2_(1,i+2)=dU2(1,i+1);

dU_(:,i+2)=[dU1_(:,i+2);dU2_(:,i+2)];

u1=U1(1,i+1);
u2=U2(1,i+1);
sim('Model')
%d(i+1)=yl(end)-Y_m(1,i); %linear
d(i+1)=y(end)-Y_m(1,i);
%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];
x01=x1(end);
x02=x2(end);

```

```

end
figure(1);
hold on
plot(y1,'m');
figure(2);
hold on
plot(u_1,'m');
figure(3);
hold on
plot(u_2,'m');
% .....
% 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];
Kdmc=(G'*Q*G+R)\(G'*Q);
alpha=0.5;
x01=0.0882;
x02=441.2;
% .....
U1_ = zeros(P,length(t));
U2_ = zeros(P,length(t));
dU1_=zeros(N-1,length(t));
dU2_=zeros(N-1,length(t));
dU_=[dU1_;dU2_];
% U_=[U1_ ; U2_];
Ud1=zeros(N+P-1,length(t));
Ud2=zeros(N+P-1,length(t));
d=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));
U1=zeros(M,length(t));
U2=zeros(M,length(t));
% U=[U1;U2];
dU1=zeros(M,length(t));
dU2=zeros(M,length(t));
dU=[dU1;dU2];
% .....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)=G_*dU(:,i+1)+g1(N+1)*U1(:,i+1)+g2(N+1)*U2(:,i+1);  
D(:,i+1)=d(i+1)*ones(P,1);
```

```
E(:,i+1)=Y_d(:,i+1)-Y_past(:,i+1)-D(:,i+1);  
dU(:,i+1)=Kdmc*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);
```

```
Ud1(2:N+P-1,i+2)=Ud1(1:N+P-2,i+1);  
Ud1(1,i+2)=U1(1,i+1);  
U1_(:,i+2)=Ud1(N:N+P-1,i+2);
```

```
Ud2(2:N+P-1,i+2)=Ud2(1:N+P-2,i+1);  
Ud2(1,i+2)=U2(1,i+1);  
U2_(:,i+2)=Ud2(N:N+P-1,i+2);
```

```
dU1_(2:N-1,i+2) = dU1_(1:N-2,i+1);  
dU1_(1,i+2)=dU1(1,i+1);  
dU2_(2:N-1,i+2) = dU2_(1:N-2,i+1);  
dU2_(1,i+2)=dU2(1,i+1);
```

```
dU_(:,i+2)=[dU1_(:,i+2);dU2_(:,i+2)];
```

```
u1=U1(1,i+1);  
u2=U2(1,i+1);  
sim('Model')  
%d(i+1)=yl(end)-Y_m(1,i); %linear  
d(i+1)=y(end)-Y_m(1,i);  
%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];  
x01=x1(end);  
x02=x2(end);  
end  
figure(1);  
hold on  
plot(y1,'c');  
hold on  
plot(r+441.22,'r')  
legend('Q=I','Q=2I','Q=60I','r');  
figure(2);
```

```

hold on
plot(u_1,'c');
legend('Q=I','Q=2I','Q=60I');
figure(3);
hold on
plot(u_2,'c');
legend('Q=I','Q=2I','Q=60I');
% .....
% % .....
% .....verifying parameter.....
% verify 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:5;
[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=P2;
N=N1;
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];
% .....Hankel Matrix.....
c1= [g1(3:P+2)];
r1 = [(g1(P+2:N+1))' zeros(1,P-1)];
G1_ = hankel(c1,r1);

```

```

% .....
c2 = [g2(3:P+2)];
r2 = [(g2(P+2:N+1))' zeros(1,P-1)];
G2_ = hankel(c2,r2);
G_=[G1_ G2_];
% .....
%% ..... Designing .....
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];
Kdmc=(G'*Q*G+R)\(G'*Q);
alpha=0.5;
x01=0.0882;
x02=441.2;
% .....
U1_ = zeros(P,length(t));
U2_ = zeros(P,length(t));
dU1_=zeros(N-1,length(t));
dU2_=zeros(N-1,length(t));
dU_=[dU1_;dU2_];
% U_=[U1_ ; U2_];
Ud1=zeros(N+P-1,length(t));
Ud2=zeros(N+P-1,length(t));
d=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));
U1=zeros(M,length(t));
U2=zeros(M,length(t));
% U=[U1;U2];
dU1=zeros(M,length(t));
dU2=zeros(M,length(t));
dU=[dU1;dU2];
% .....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)=G_*dU(:,i+1)+g1(N+1)*U1_(:,i+1)+g2(N+1)*U2_(:,i+1);
D(:,i+1)=d(i+1)*ones(P,1);

```

```

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

```

```

Ud1(2:N+P-1,i+2)=Ud1(1:N+P-2,i+1);
Ud1(1,i+2)=U1(1,i+1);
U1_(:,i+2)=Ud1(N:N+P-1,i+2);

```

```

Ud2(2:N+P-1,i+2)=Ud2(1:N+P-2,i+1);
Ud2(1,i+2)=U2(1,i+1);
U2_(:,i+2)=Ud2(N:N+P-1,i+2);

```

```

dU1_(2:N-1,i+2) = dU1_(1:N-2,i+1);
dU1_(1,i+2)=dU1(1,i+1);
dU2_(2:N-1,i+2) = dU2_(1:N-2,i+1);
dU2_(1,i+2)=dU2(1,i+1);

```

```

dU_(:,i+2)=[dU1_(:,i+2);dU2_(:,i+2)];

```

```

u1=U1(1,i+1);
u2=U2(1,i+1);
sim('Model')
% d(i+1)=yl(end)-Y_m(1,i); % linear
d(i+1)=y(end)-Y_m(1,i);
% 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];
x01=x1(end);
x02=x2(end);
end
figure(1);
plot(y1,'b');
grid on
title('Response of the nonlinear system');
xlabel('sample');
figure(2);
plot(u_1,'b');
grid on
xlabel('sample');
title('Control law for input 1 without bias');
figure(3);

```

```

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: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=P2;
N=N1;
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];
%.....Hankel Matrix.....
c1= [g1(3:P+2)];
r1 = [(g1(P+2:N+1))' zeros(1,P-1)];
G1_ = hankel(c1,r1);
%.....
c2 = [g2(3:P+2)];
r2 = [(g2(P+2:N+1))' zeros(1,P-1)];
G2_ = hankel(c2,r2);
G_=[G1_ G2_];
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];
Kdmc=(G'*Q*G+R)\(G'*Q);
alpha=0.5;
x01=0.0882;
x02=441.2;
% .....
U1_ = zeros(P,length(t));
U2_ = zeros(P,length(t));
dU1_=zeros(N-1,length(t));
dU2_=zeros(N-1,length(t));
dU_=[dU1_;dU2_];
% U_=[U1_ ; U2_];
Ud1=zeros(N+P-1,length(t));
Ud2=zeros(N+P-1,length(t));
d=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));
U1=zeros(M,length(t));
U2=zeros(M,length(t));
% U=[U1;U2];
dU1=zeros(M,length(t));
dU2=zeros(M,length(t));
dU=[dU1;dU2];
% .....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)=G_*dU(:,i+1)+g1(N+1)*U1(:,i+1)+g2(N+1)*U2(:,i+1);
D(:,i+1)=d(i+1)*ones(P,1);

E(:,i+1)=Y_d(:,i+1)-Y_past(:,i+1)-D(:,i+1);
dU(:,i+1)=Kdmc*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);
```

```
Ud1(2:N+P-1,i+2)=Ud1(1:N+P-2,i+1);
Ud1(1,i+2)=U1(1,i+1);
U1_(:,i+2)=Ud1(N:N+P-1,i+2);
```

```
Ud2(2:N+P-1,i+2)=Ud2(1:N+P-2,i+1);
Ud2(1,i+2)=U2(1,i+1);
U2_(:,i+2)=Ud2(N:N+P-1,i+2);
```

```
dU1_(2:N-1,i+2) = dU1_(1:N-2,i+1);
dU1_(1,i+2)=dU1(1,i+1);
dU2_(2:N-1,i+2) = dU2_(1:N-2,i+1);
dU2_(1,i+2)=dU2(1,i+1);
```

```
dU_(:,i+2)=[dU1_(:,i+2);dU2_(:,i+2)];
```

```
u1=U1(1,i+1);
u2=U2(1,i+1);
sim('Model')
% d(i+1)=yl(end)-Y_m(1,i); % linear
d(i+1)=y(end)-Y_m(1,i);
% 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];
x01=x1(end);
x02=x2(end);
end
figure(1);
hold on
plot(y1,'m');
figure(2);
hold on
plot(u_1,'m');
figure(3);
hold on
plot(u_2,'m');
% .....
% 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:50;
[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=P2;
N=N1;
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];
% .....Hankel Matrix.....
c1= [g1(3:P+2)];
r1 = [(g1(P+2:N+1))' zeros(1,P-1)];
G1_ = hankel(c1,r1);
% .....
c2 = [g2(3:P+2)];
r2 = [(g2(P+2:N+1))' zeros(1,P-1)];
G2_ = hankel(c2,r2);
G_=[G1_ G2_];
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];
Kdmc=(G'*Q*G+R)\(G'*Q);
alpha=0.5;
x01=0.0882;
x02=441.2;
% .....
U1_ = zeros(P,length(t));
U2_ = zeros(P,length(t));
dU1_=zeros(N-1,length(t));

```

```

dU2_=zeros(N-1,length(t));
dU_=[dU1_;dU2_];
% U_=[U1_ ; U2_];
Ud1=zeros(N+P-1,length(t));
Ud2=zeros(N+P-1,length(t));
d=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));
U1=zeros(M,length(t));
U2=zeros(M,length(t));
% U=[U1;U2];
dU1=zeros(M,length(t));
dU2=zeros(M,length(t));
dU=[dU1;dU2];
% .....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)=G_*dU(:,i+1)+g1(N+1)*U1_(:,i+1)+g2(N+1)*U2_(:,i+1);
D(:,i+1)=d(i+1)*ones(P,1);

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

Ud1(2:N+P-1,i+2)=Ud1(1:N+P-2,i+1);
Ud1(1,i+2)=U1(1,i+1);
U1_(:,i+2)=Ud1(N:N+P-1,i+2);

Ud2(2:N+P-1,i+2)=Ud2(1:N+P-2,i+1);
Ud2(1,i+2)=U2(1,i+1);
U2_(:,i+2)=Ud2(N:N+P-1,i+2);

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

```

```
dU1_(1,i+2)=dU1(1,i+1);
dU2_(2:N-1,i+2) = dU2_(1:N-2,i+1);
dU2_(1,i+2)=dU2(1,i+1);
```

```
dU_(:,i+2)=[dU1_(:,i+2);dU2_(:,i+2)];
```

```
u1=U1(1,i+1);
u2=U2(1,i+1);
sim('Model')
%d(i+1)=yl(end)-Y_m(1,i); %linear
d(i+1)=y(end)-Y_m(1,i);
%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];
x01=x1(end);
x02=x2(end);
end
figure(1);
hold on
plot(y1,'c');
hold on
plot(r+441.22,'r')
legend('Ts=0.05','Ts=0.1','Ts=0.5','r');
figure(2);
hold on
plot(u_1,'c');
legend('Ts=0.05','Ts=0.1','Ts=0.5');
figure(3);
hold on
plot(u_2,'c');
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]=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=P2;
N=N1;
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];
%.....Hankel Matrix.....
c1= [g1(3:P+2)];
r1 = [(g1(P+2:N+1))' zeros(1,P-1)];
G1_ = hankel(c1,r1);
%.....
c2 = [g2(3:P+2)];
r2 = [(g2(P+2:N+1))' zeros(1,P-1)];
G2_ = hankel(c2,r2);
G_=[G1_ G2_];
%.....
%%..... Designing.....
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];
Kdmc=(G'*Q*G+R)\(G'*Q);
alpha=0.5;
x01=0.0882;
x02=441.2;
%.....
U1_ = zeros(P,length(t));
U2_ = zeros(P,length(t));
dU1_=zeros(N-1,length(t));
dU2_=zeros(N-1,length(t));
dU_=[dU1_;dU2_];
% U_=[U1_ ; U2_];

```



```

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

Y_past(:,i+1)=G_*dU(:,i+1)+g1(N+1)*U1(:,i+1)+g2(N+1)*U2(:,i+1);
D(:,i+1)=d(i+1)*ones(P,1);

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

Ud1(2:N+P-1,i+2)=Ud1(1:N+P-2,i+1);
Ud1(1,i+2)=U1(1,i+1);
U1_(:,i+2)=Ud1(N:N+P-1,i+2);

Ud2(2:N+P-1,i+2)=Ud2(1:N+P-2,i+1);
Ud2(1,i+2)=U2(1,i+1);
U2_(:,i+2)=Ud2(N:N+P-1,i+2);

dU1_(2:N-1,i+2) = dU1_(1:N-2,i+1);
dU1_(1,i+2)=dU1(1,i+1);
dU2_(2:N-1,i+2) = dU2_(1:N-2,i+1);
dU2_(1,i+2)=dU2(1,i+1);

dU_(:,i+2)=[dU1_(:,i+2);dU2_(:,i+2)];

u1=U1(1,i+1);
u2=U2(1,i+1);
sim('Model')
%d(i+1)=yl(end)-Y_m(1,i); %linear
d(i+1)=y(end)-Y_m(1,i);
%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);
plot(y1,'b');
hold on

```

```

plot(r+441.22,'r');
grid on
axis([0 45 439 447]);
legend('y','r');
title('Response of the nonlinear system');
xlabel('sample');
figure(4);
plot(y1-441.22,'b');
hold on
plot(ym,'r');
grid on
xlabel('sample');
title('Ym and Yp without bias');
legend('YPlant','YModel');
figure(5);
plot(u_1,'b');
grid on
xlabel('sample');
title('Control law for input 1 without bias');
figure(6);
plot(u_2,'b');
grid on
xlabel('sample');
title('Control law for input 2 without bias');
% figure(7);
% plot(dist,'b');
% grid on
% xlabel('sample');
% title('disturbance');
% figure(7);
% plot(noise,'b');
% grid on
% xlabel('sample');
% title('white noise');

```

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.

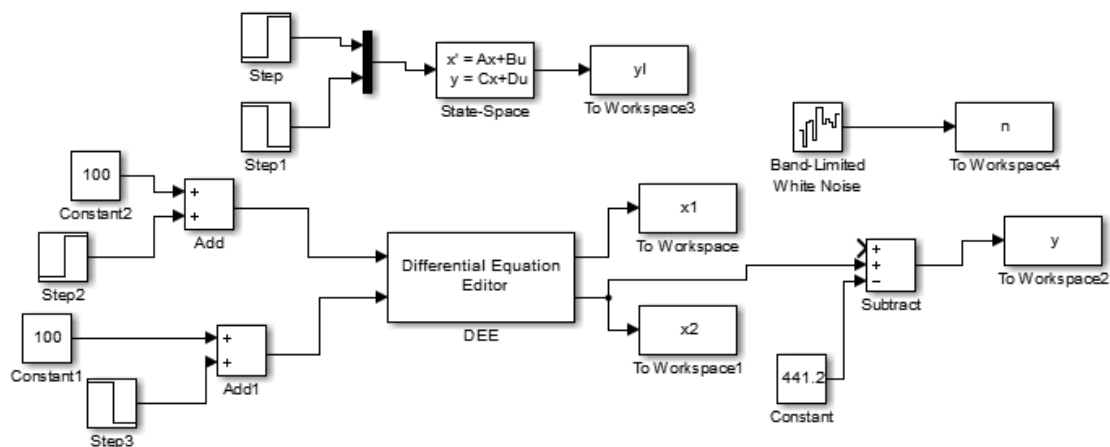


Figure 18. Simulation of the linear and nonlinear plant in Simulink.

کد و شبیه سازی زیر به منظور ایجاد g ها توسط سیستم غیر خطی نوشته شده است.

```
Ts=0.1;
n=1;
g1=[];
g2=[];
u_0=[];
for i=0.1:0.1:10
    c=i;
    if c<=2
        u0=0;
    elseif c<=4 && c>2
        u0=1;
    elseif c<=6 && c>4
        u0=0;
    elseif c<=8 && c>6
        u0=-1;
    elseif c<=10 && c>8
        u0=0;
    end
    sim('gs');
    g1=[g1; g_1];
    g2=[g2; g_2];
    u_0=[u_0; u0];
end
%.....input.....
% g1=[];
% g2=[];
% u_0=[];
% for i=0.1:0.1:10
%     c=i;
%     if c<=2
%         u0=0;
%     elseif c<=4 && c>2
%         u0=1;
%     elseif c<=6 && c>4
%         u0=0;
%     elseif c<=8 && c>6
%         u0=-1;
%     elseif c<=10 && c>8
%         u0=0;
%     end
%     sim('gs');
%     g1=[g1; g_1(end)];
%     g2=[g2; g_2(end)];
%     u_0=[u_0; u0];
% end
%.....
% g11=[];
% g21=[];
% u_0=[];
% for i=0.1:0.1:10
%     c=i;
%     if c<=2
%         u0=0;
%     elseif c<=10 && c>2
%         u0=1;
```

```

% end
% sim('gs');
% g11=[g11; g_1(end)];
% g21=[g21; g_2(end)];
% u_0=[u_0; u0];
% end
% %.....
% g12=[];
% g22=[];
% u_02=[];
% for i=0.1:0.1:10
%     c=i;
%     if c<=4
%         u0=0;
%     elseif c<=10 && c>4
%         u0=-1;
%     end
%     sim('gs');
%     g12=[g12; g_1(end)];
%     g22=[g22; g_2(end)];
%     u_02=[u_02; u0];
% end
% %.....
% g13=[];
% g23=[];
% u_03=[];
% for i=0.1:0.1:10
%     c=i;
%     if c<=6
%         u0=0;
%     elseif c<=10 && c>6
%         u0=1;
%     end
%     sim('gs');
%     g13=[g13; g_1(end)];
%     g23=[g23; g_2(end)];
%     u_03=[u_03; u0];
% end
% %.....
% g14=[];
% g24=[];
% u_04=[];
% for i=0.1:0.1:10
%     c=i;
%     if c<=8
%         u0=0;
%     elseif c<=10 && c>8
%         u0=-1;
%     end
%     sim('gs');
%     g14=[g14; g_1(end)];
%     g24=[g24; g_2(end)];
%     u_04=[u_04; u0];
% end
% g1=(g11+g12+g13+g14)/4;
% g2=(g21+g22+g23+g24)/4;

```

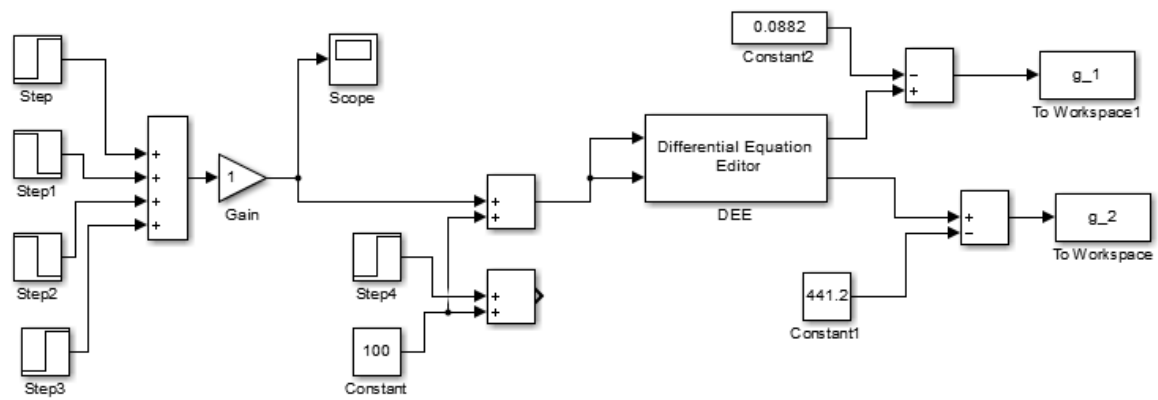


Figure 19. Simulation for producing g_i in Simulink.