



04.1395

پروژه ششم درس MPC

طراحی کنترل کننده EDMC

نام دانشجو: فاطمه زاهدی

شماره دانشجویی: 94201583

نام استاد: دکتر حائری

فهرست:

1. معرفی سیستم مورد استفاده: 2
2. طراحی کنترل کننده EDMC: 2
 - 2.1 EDMC بدون محدودیت: 2
 - 2.2 EDMC با وجود محدودیت: 3
3. پاسخ سیستم به ازای خروجی مطلوب‌های مختلف: 5
 - 3.1 خروجی مطلوب سینوس: 5
 - 3.2 خروجی مطلوب پله‌ای با پرش‌های مختلف: 6
 - 3.3 خروجی مطلوب موج مربعی: 7
4. بررسی اثر پارامترها: 9
 - 4.1 بررسی ماتریس R: 9
 - 4.2 بررسی ماتریس Q: 10
 - 4.3 بررسی پارامتر آلفا: 11
 - 4.4 بررسی پارامتر بتا: 11
 - 4.5 بررسی پارامتر P: 12
 - 4.6 بررسی پارامتر M: 13
 - 4.7 بررسی پارامتر Ts: 13
5. بررسی اثر اغتشاش و نویز: 14
 - 5.1 بررسی اثر اغتشاش: 14
 - 5.2 بررسی اثر نویز: 15
6. بررسی عدم قطعیت در پروسه: 16
7. کدها و شبیه‌سازی‌ها: 17

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.

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

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

خروجی این سیستم برابر با $Y=T$ می باشد.

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 EDMC:

در این پروژه هدف طراحی کنترل کننده EDMC است که از مدل لحظه ای برای این طراحی استفاده شده است. ابتدا مقادیر P , M , β , δ و T_s را مشخص می کنیم. مقادیر P و T_s مشابه کاری که در پروژه های قبل انجام دادیم محاسبه می کنیم و مقدار $P=5$ و مقدار $T_s=0.1$ بدست آمد. ما مقدار M را نیز برابر با 3 قرار داده ایم. مقدار β را 0.4 گذاشته و مقدار δ که بایستی یک مقدار کوچک باشد را برابر با 0.01 قرار داده ایم. بر اساس این مقادیر طراحی این کنترل کننده را با وجود محدودیت و بدون محدودیت انجام داده و نتایج را در بخش های بعد آورده ایم.

2.1 EDMC بدون محدودیت:

بعد از طراحی کنترل کننده پاسخ سیستم به ازای پله واحد با وجود این کنترل کننده در زیر آورده شده است. علاوه بر پاسخ سیستم قانون های کنترلی هر دو ورودی در زیر رسم شده است.

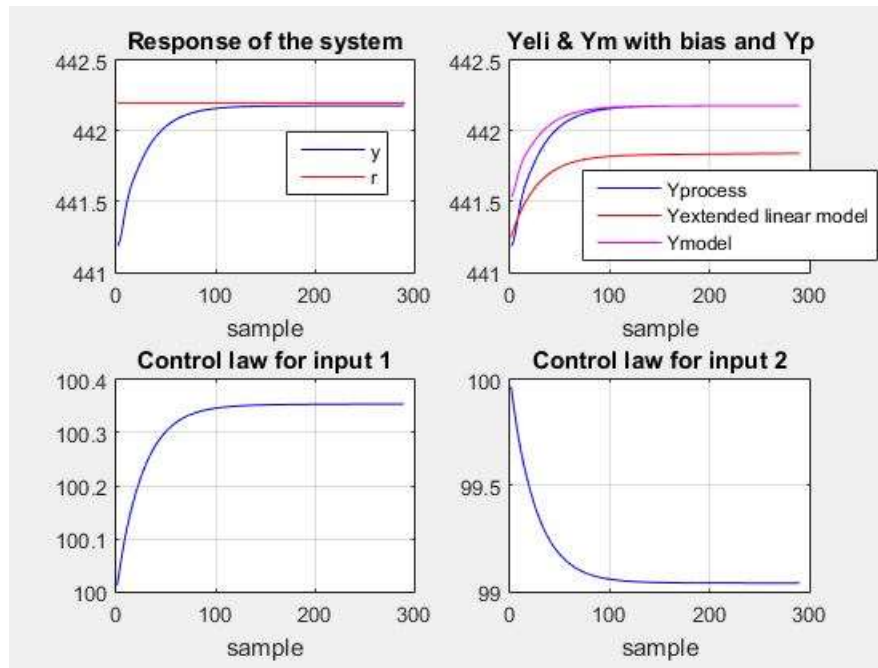


Figure1. EDMC on nonlinear system without constraint.

2.2 EDMC با وجود محدودیت:

در این بخش با وجود محدودیت روی ورودی‌ها و حالات و خروجی این کنترل کننده را طراحی می‌کنیم. محدودیت‌های مذکور به صورت زیر می‌باشد.

$$100.1 \leq u_1 \leq 100.4, 99.3 \leq u_2 \leq 99.8, 0.0855 \leq x_1 \leq 0.08742, 441.44 \leq x_2 \leq 442, y = x_2.$$

نحوه بدست آوردن این محدودیت‌ها بدین صورت می‌باشد که با توجه به اینکه بایستی محدودیت‌های موثر انتخاب شود ابتدا به محدوده تغییرات هر کدام در حالت بدون محدودیت توجه می‌کنیم و با توجه به آن محدوده محدوده‌هایی را که در بالا آورده شده است مشخص می‌کنیم. اما برای اعمال این محدودیت‌ها به کنترل کننده مان بایستی کلیه محدودیت‌ها به محدوده‌ای بر حسب u تبدیل شود. این کار به این علت است که تابع هدف ما بر حسب u می‌باشد و با این کار یکی از قیود که دینامیک سیستم هست نیز در بدست آوردن قانون کنترلی موثر واقع می‌شود.

بعد از پیاده سازی این کنترل کننده با وجود محدودیت‌های مذکور نتایج به صورت زیر شد. در این قسمت به منظور مشاهده تاثیر این محدودیت‌ها شکل حالت اول نیز آورده شده است حالت دوم نیز که همان خروجی می‌باشد.

در شکل بالا به وضوح می‌توان تاثیر محدودیت‌ها را با برش‌هایی که اتفاق افتاده است دید.

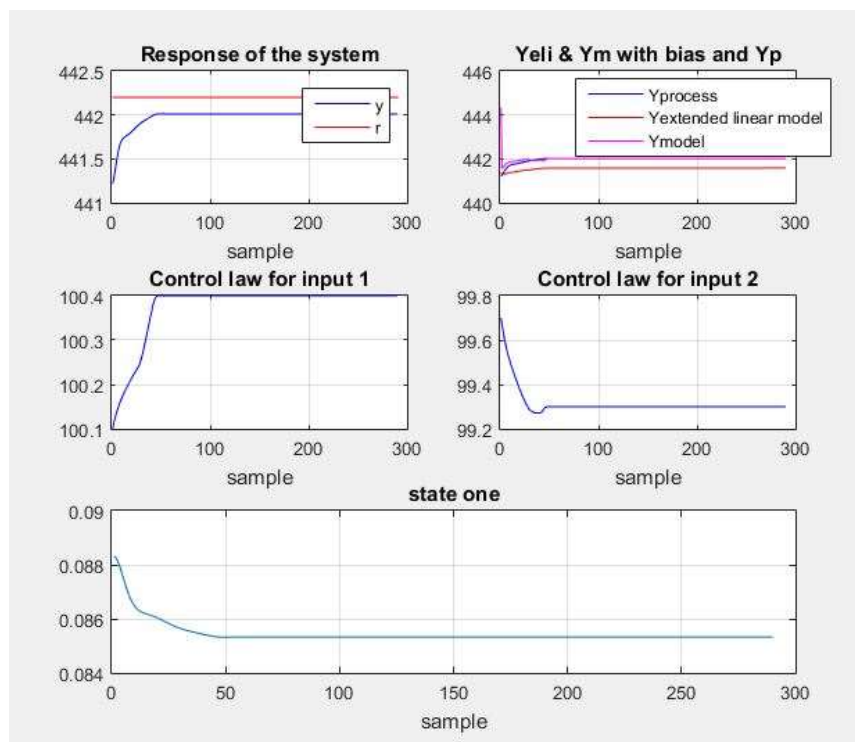


Figure2. EDMC on nonlinear system with constraint.

به منظور اینکه تاثیر سایر حد بالاها و حد پايين هايي که در اين شکل خودشان را بخاطر نوع خروجی مطلوب نشان نداده اند ببينيم، ما پاسخ سيستم به ازای خروجی مطلوب سينوسی را نيز رسم کرده ایم که در شکل زیر آورده شده است.

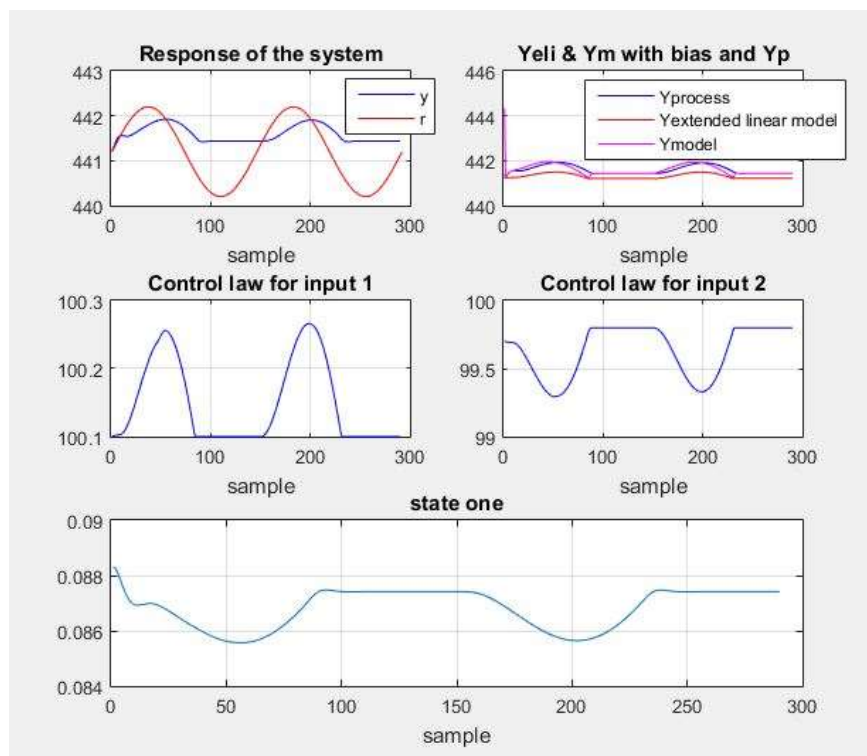


Figure3. EDMC on nonlinear system with constraint with sin wave as set point.

در شکل بالا کاملاً تاثیر محدودیت‌ها را می‌توان دید و با دقت به آن‌ها می‌بینیم که دقیقاً در همان محدوده‌های مذکور باند شده‌اند و برش خورده‌اند.

3. پاسخ سیستم به ازای خروجی مطلوب‌های مختلف:

در این بخش خروجی‌های مطلوب مختلفی را به سیستم می‌دهیم و پاسخ سیستم با وجود عدم قطعیت و بدون عدم قطعیت بررسی می‌کنیم. لازم به ذکر است با توجه به اینکه محدوده‌های ذکر شده در بالا محدوده‌های بسیار کوچکی هستند و نمی‌توان به خوبی تاثیر پاسخ مطلوب را دید ما تنها در این بخش محدوده‌های بزرگتری را نسبت به حالت قبل داده‌ایم که باندهای آن در کدها آمده است.

3.1. خروجی مطلوب سینوس:

در این بخش به ازای خروجی مطلوب سینوسی پاسخ سیستم را بررسی می‌کنیم. با توجه به خواسته مساله ابتدا فرکانس مناسبی را که در آن می‌توان دنبال کردن خروجی مطلوب به به وضوح دید انتخاب می‌کنیم و در آن به بررسی می‌پردازیم. در زیر پاسخ سیستم به ازای این خروجی مطلوب را در حالت بدون محدودیت می‌توان دید.

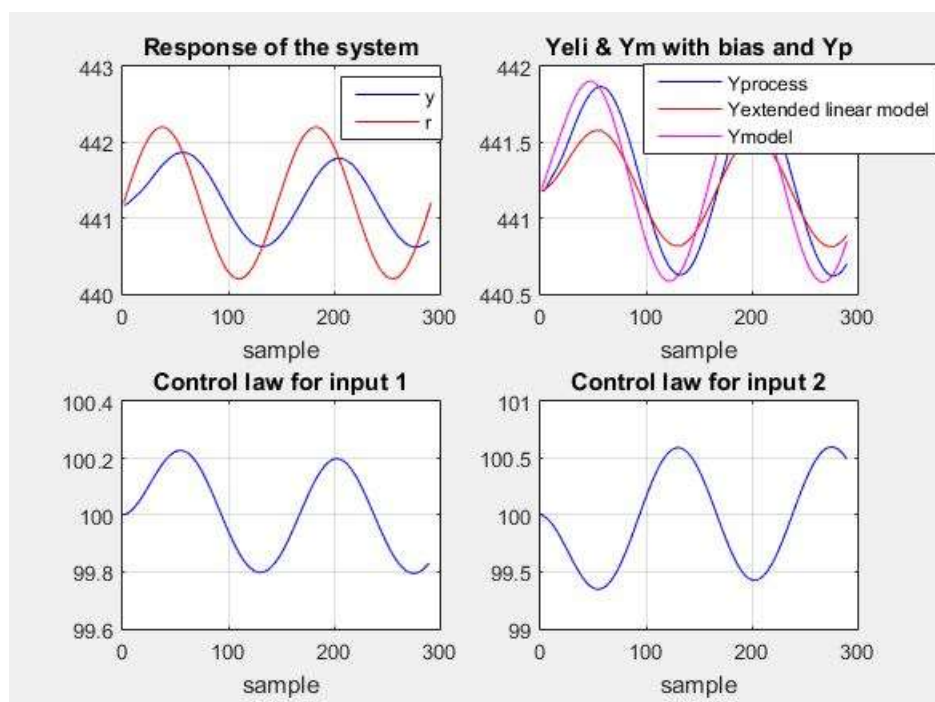


Figure 4. Response of the system with sine wave as a desired output.

همانطور که در شکل بالا می‌بینیم پاسخ سیستم خروجی مطلوب را دنبال می‌کند. علاوه بر این می‌توان تاثیر تغییر فرکانس را در این قسمت بررسی کرد اما با توجه به اینکه بررسی فرکانس خواسته نشده است از بررسی آن صرف نظر می‌کنیم.

3.2. 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. در زیر حالت بدون محدودیت آورده شده است.

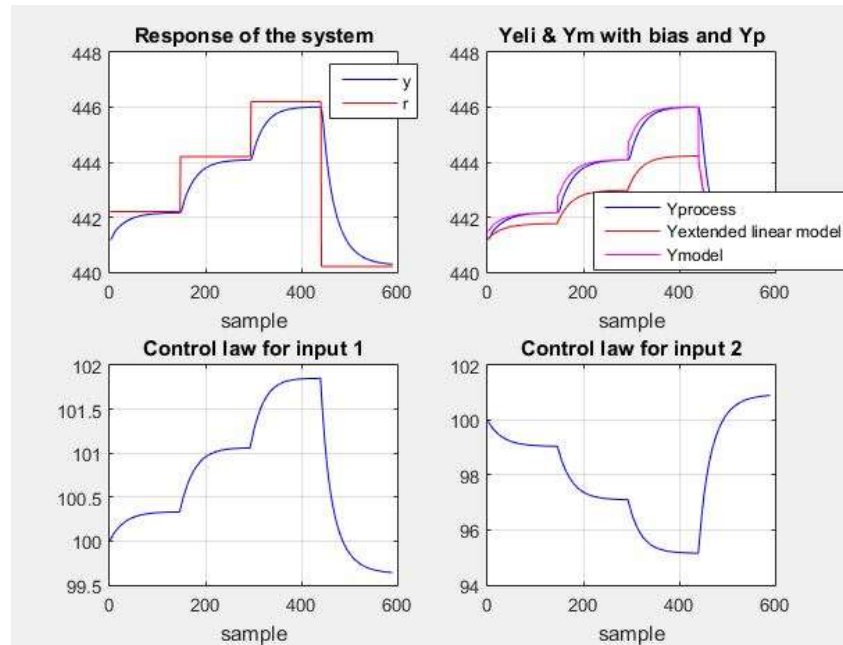


Figure 5. Response of the system with step with various jumps as a desired output.

پاسخ به ازای این خروجی مطلوب با وجود محدودیت در شکل زیر آورده شده است.

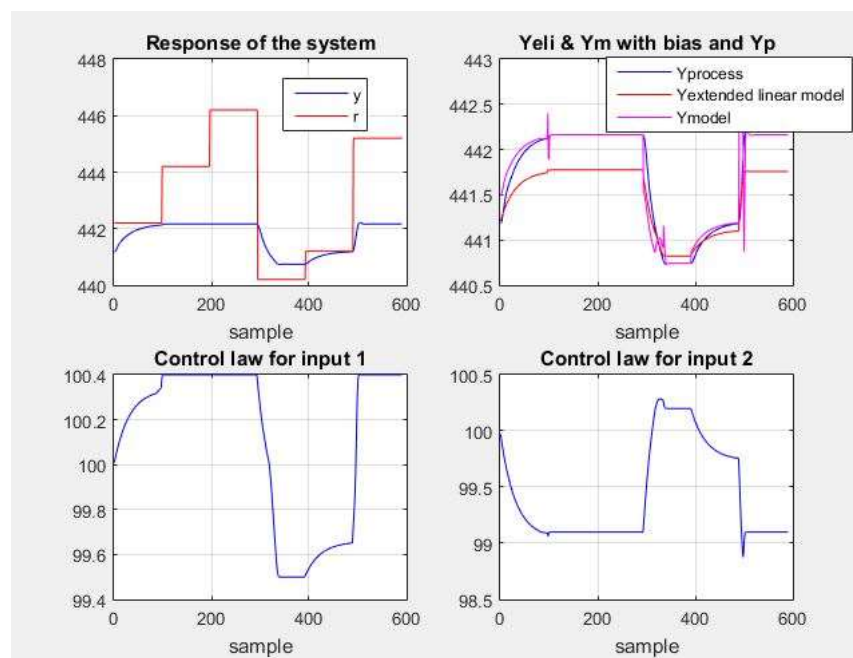


Figure 6. Response of the system with step with various jumps as a desired output with constraint.

3.3. خروجی مطلوب موج مویی:

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 0.1%، 1% و 3% نقطه کار در حالت بدون محدودیت رسم می کنیم. شکل زیر پاسخ را به ازای این خروجی مطلوب با دامنه 0.1% نقطه کار نشان می دهد.

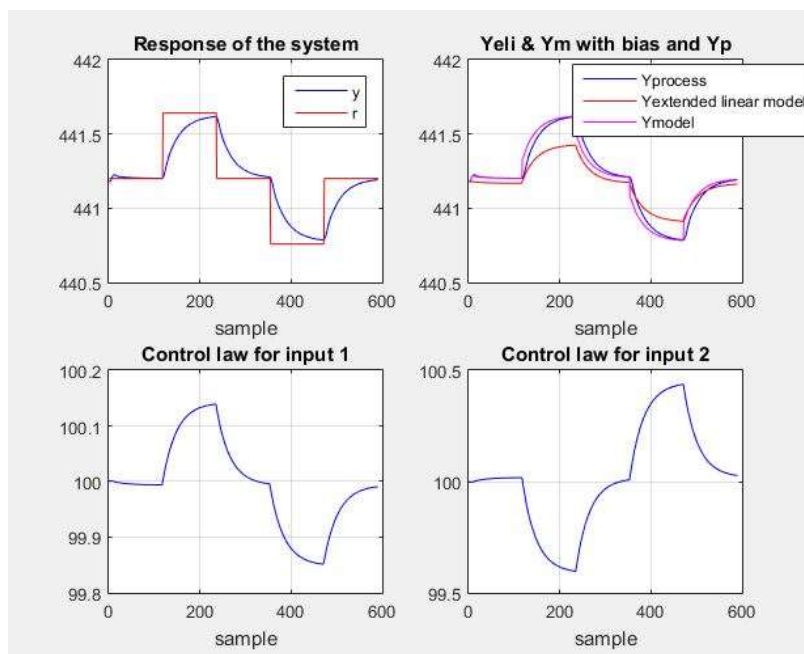


Figure 7. Response of the system with step with 0.1% of operating point as a desired output.

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

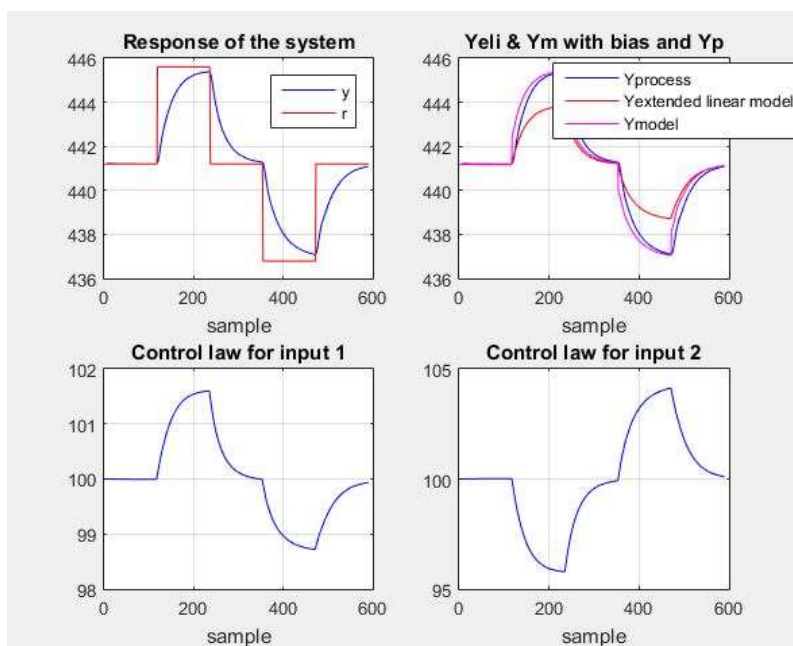


Figure 8. Response of the system with step with 1% of operating point as a desired output.

همانطور که در دو 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.

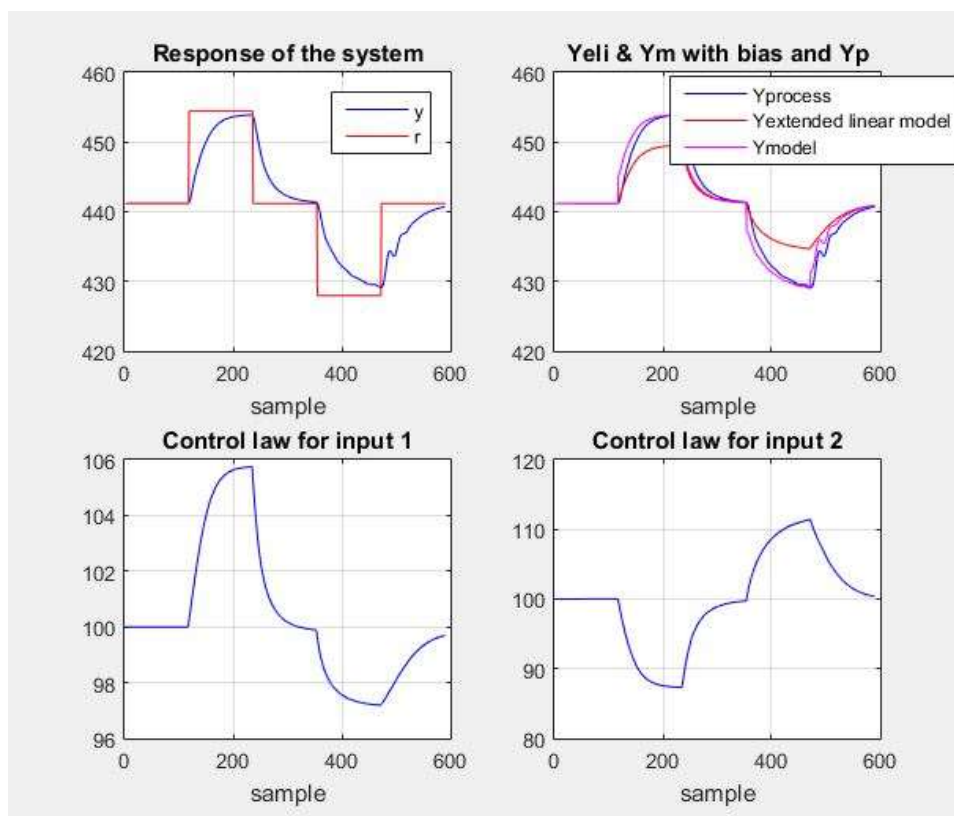


Figure 9. Response of the system with step with 3% of operating point as a desired output.

همانطور که در شکل بالا می‌بینیم پاسخ سیستم خروجی مطلوب را دنبال کرده است. از طرفی می‌بینیم که در ناحیه منفی پاسخ سیستم خیلی کمتر از کنترل‌کننده‌های قبلی در این درصد از نقطه کار به نوسان افتاده است که این برتری این کنترل‌کننده را نسبت به کنترل‌کننده‌های قبلی در بهتر بودن پاسخ نشان می‌دهد.

پاسخ سیستم به این نوع خروجی مطلوب با وجود محدودیت در شکل زیر آورده شده است.

همانطور که در این شکل می‌بینیم در ناحیه‌هایی که خروجی در بازه‌ی محدودیت ما است خروجی مطلوب را دنبال کرده است اما در نواحی که از محدوده خارج شده است در آن محدوده باند شده است.

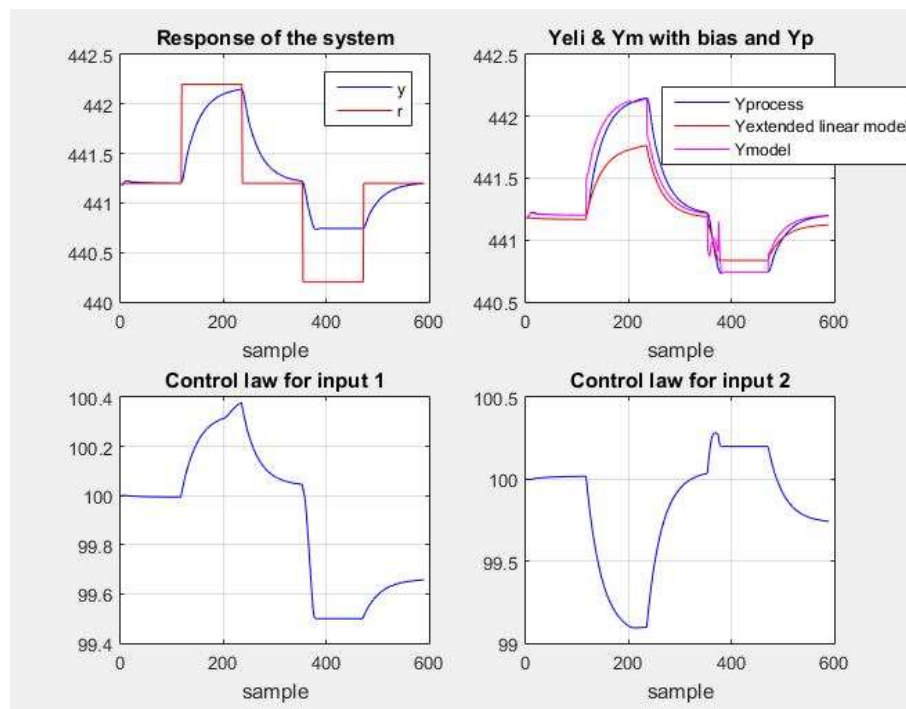


Figure 10. Response of the system as a desired output with constraint.

4. بررسی اثر پارامترها:

در این بخش تاثیر تغییرات پارامترهای مورد استفاده در طراحی کنترلر EDMC را بر روی پاسخ سیستم بررسی می کنیم. لازم به ذکر است که به منظور اینکه تاثیر پارامترها را به درستی ببینیم و تاثیر محدودیت در بررسی این تاثیر اشتباه نشود در این بخش این بررسی بدون محدودیت انجام شده است.

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 1/60 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. همانطور که در شکل 11 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.

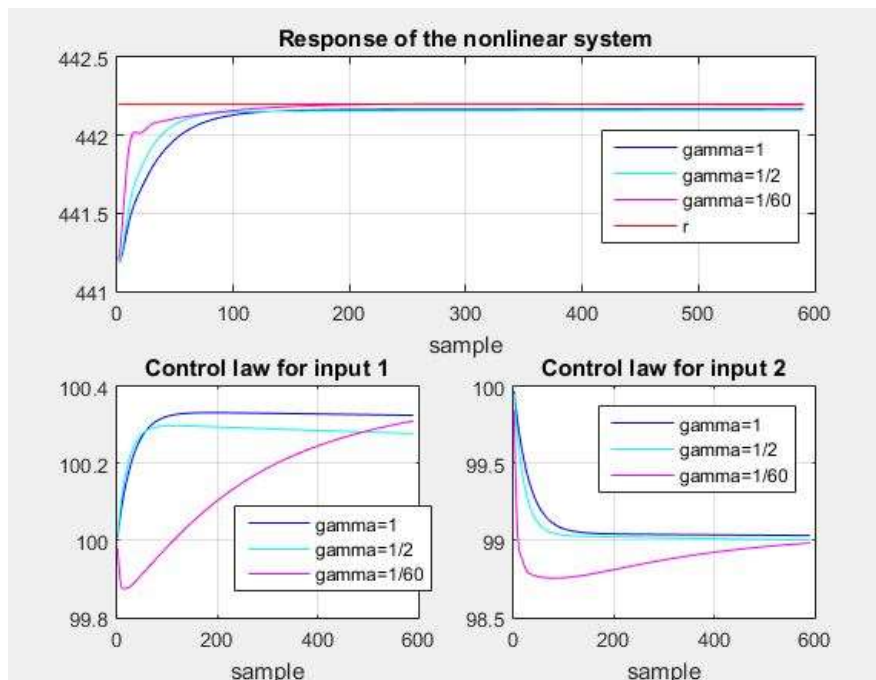


Figure 11. Responses of the system in verifying R .

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

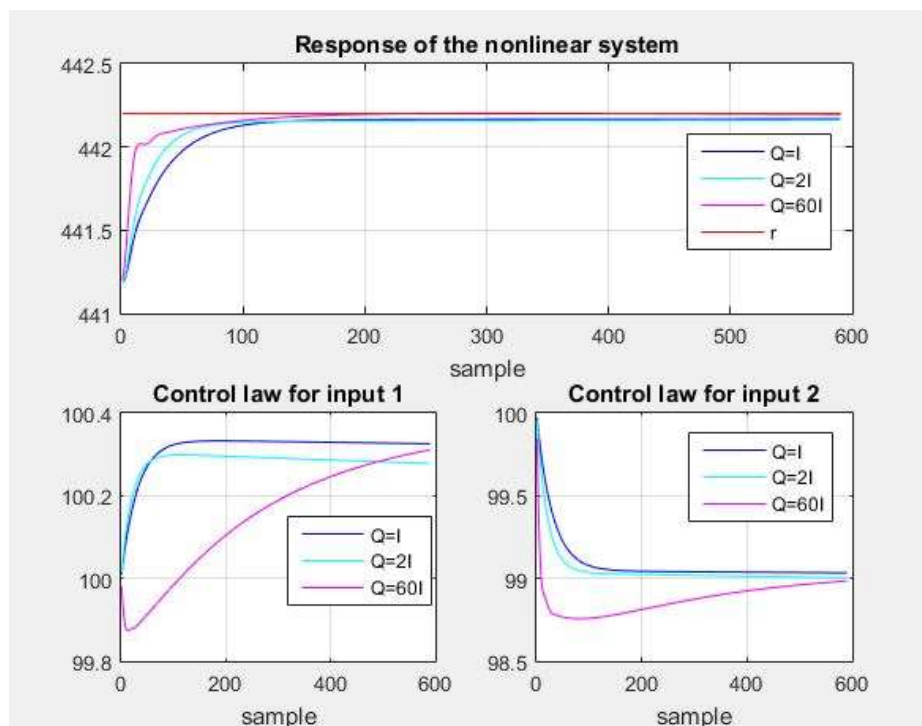


Figure 12. Responses of the system in verifying Q .

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

4.3. بررسی پارامتر آلفا:

در این قسمت تاثیر پارامتر آلفا را بر روی پاسخ سیستم بررسی می‌کنیم. شکل زیر پاسخ سیستم به ازای مقادیر آلفا 0.1، 0.5 و 0.8 نشان می‌دهد. در طی این بررسی مقدار گاما یک قرار داده شده است.

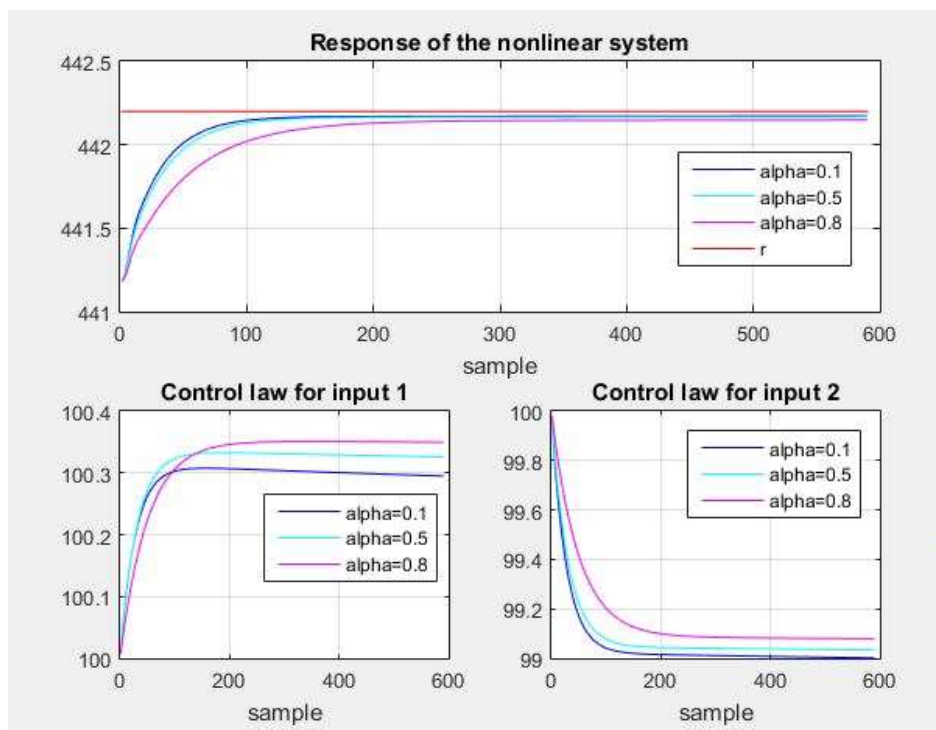


Figure 13. Responses of the system in verifying α .

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

4.4. بررسی پارامتر بتا:

در این بخش به بررسی تاثیر تغییرات پارامتر بتا در پاسخ سیستم می‌پردازیم. بتا پارامتری است که در بدست آوردن D^{nl} مورد استفاده قرار می‌گیرد.

همانطور که در شکل زیر می‌بینیم کاهش بتا تا حدودی باعث ایجاد خطای حالت ماندگار شده است. از طرفی کاهش گاما باعث کند شدن پاسخ سیستم شده است که این تاثیر کاملاً با توجه به این پارامتر قابل انتظار ما بود.

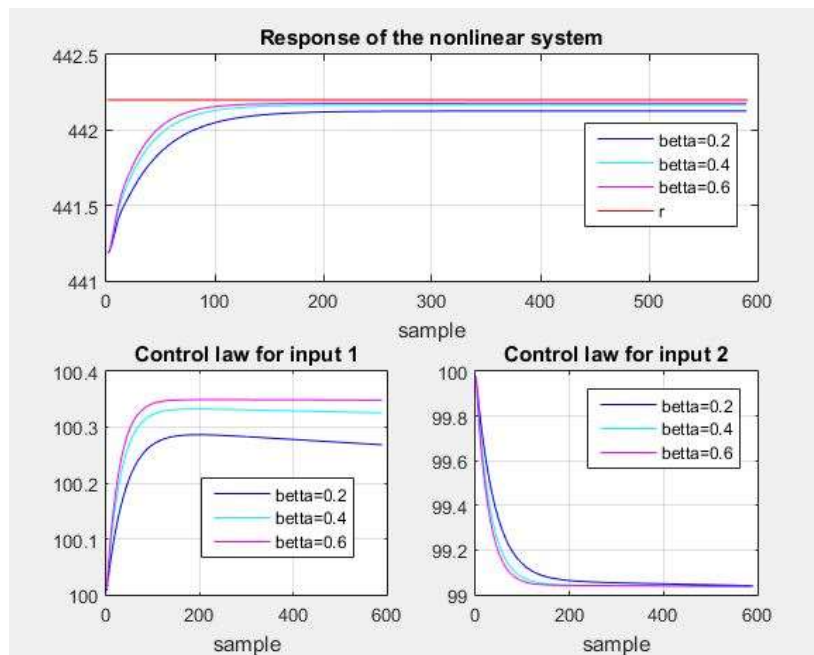


Figure 14. Responses of the system in verifying β .

4.5. بررسی پارامتر P :

در این بخش به بررسی تاثیر پارامتر P بر روی پاسخ سیستم می پردازیم. پاسخ سیستم را به ازای P های 5، 10 و 15 رسم شده است. شکل 15 پاسخ سیستم به ازای P های مختلف را نشان می دهد. لازم به ذکر است که مقدار M برابر با 3 قرار داده شده است.

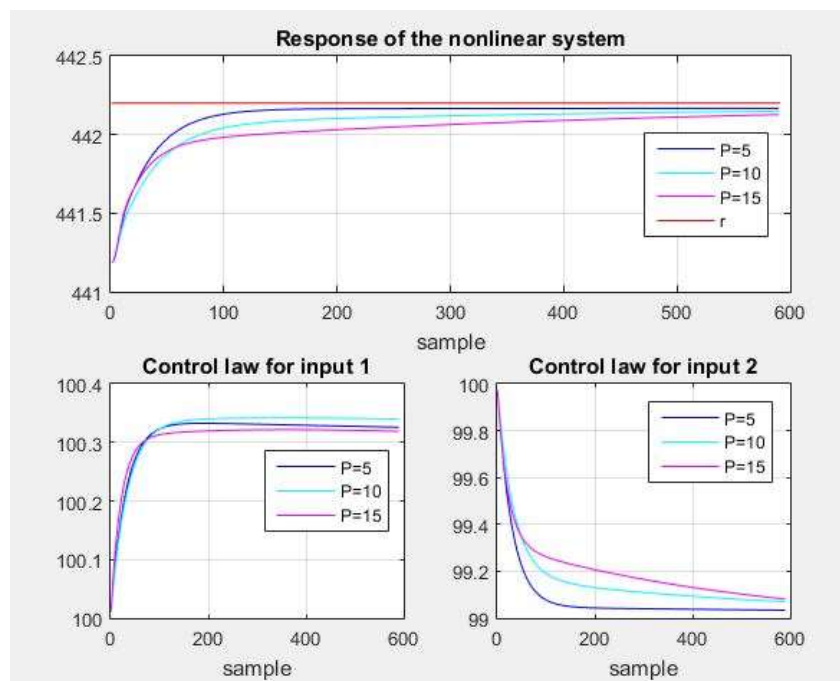


Figure 15. Responses of the system in verifying P .

می‌بینیم که افزایش P تا حدودی پاسخ را کند تر کرده است افزایش P پاسخ دقیق تری را به ما می‌دهد اما با توجه به اینکه ما در بدست آوردن خطای غیر خطی تعداد $iteration$ ها را محدود کرده‌ایم که از 20 مرحله بیشتر نشود این کندی در پاسخ را می‌بینیم.

4.6. بررسی پارامتر M :

در این بخش به بررسی تاثیر پارامتر M بر روی پاسخ سیستم می‌پردازیم. پاسخ سیستم را به ازای M های 1، 3 و 5 رسم شده است. شکل 16 پاسخ سیستم به ازای M های مختلف را نشان می‌دهد. لازم به ذکر است که مقدار P برابر با 5 قرار داده شده است.

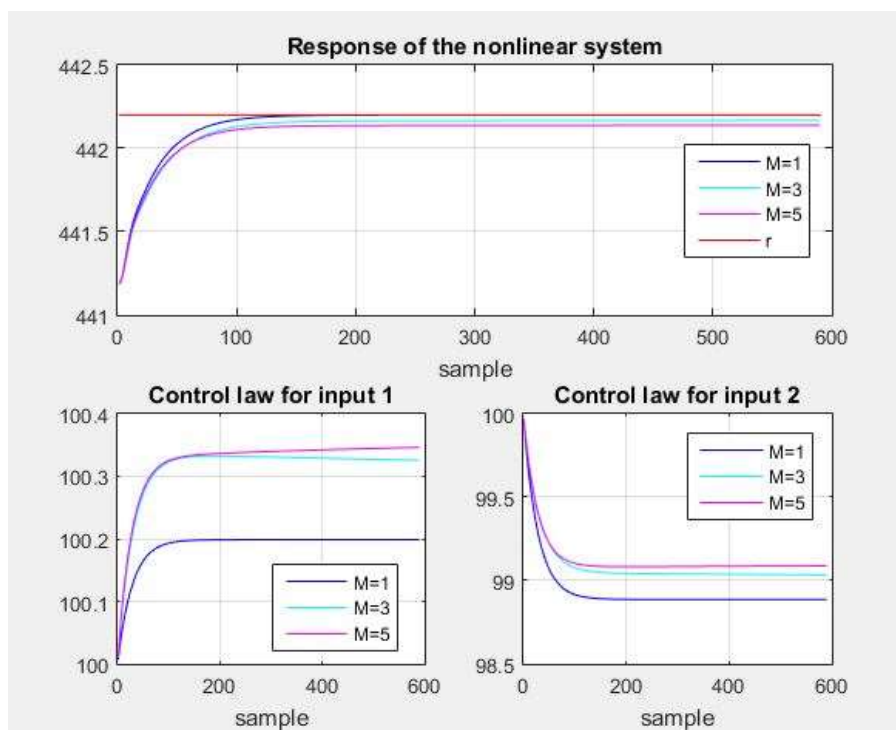


Figure 16. Responses of the system in verifying M .

تاثیری مشابه تاثیر P را برای بررسی M می‌توانیم ببینیم.

4.7. 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.

همانطور که در شکل زیر می‌بینیم وقتی زمان نمونه برداری را کم می‌کنیم باعث تند شدن پاسخ سیستم می‌شود. اما این تند شدن پاسخ تا جایی ادامه دارد یعنی از یک مقداری به بعد دیگر کاهش T_s تاثیر چندانی در سریع شدن پاسخ ندارد. البته این تاثیر در این کنترل کننده اینقدر کم است که بدون زوم کردن بر تصویر به وضوح دیده نمی‌شود.

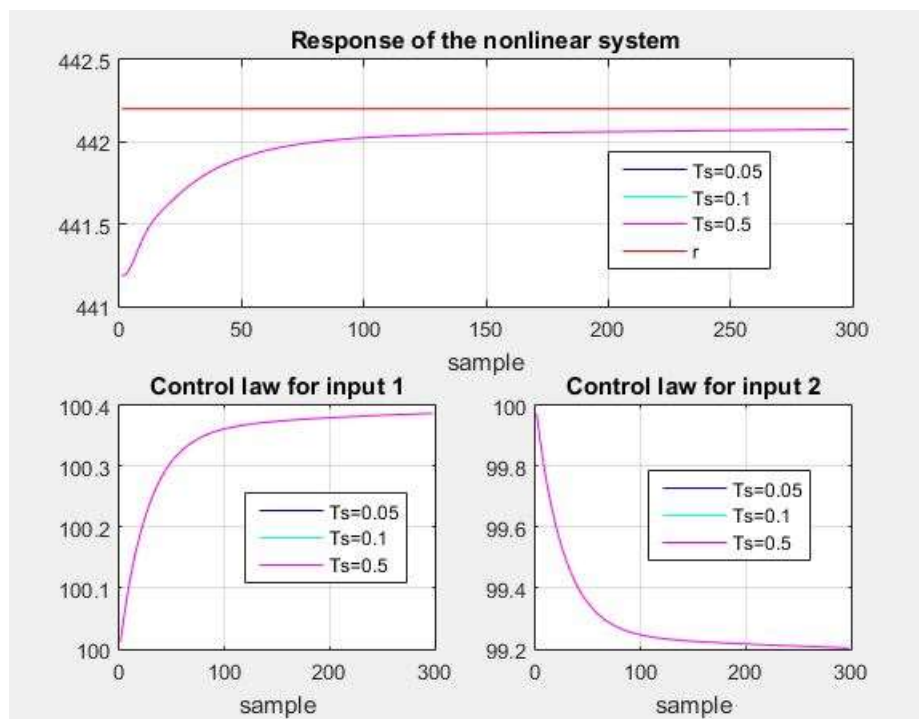


Figure 17. Responses of the system in verifying T_s .

5. بررسی اثر اغتشاش و نویز:

در این بخش روی خروجی پروسه نویز و اغتشاش اضافه می‌کنیم و اثر اضاف شدن این دو را در پاسخ سیستم مشاهده می‌کنیم. لازم به ذکر است کلیه این بررسی‌ها با وجود محدودیت صورت گرفته است.

5.1. بررسی اثر اغتشاش:

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

محدودیت‌های مربوط به این قسمت همان محدودیت‌های اصلی است که در بخش اول گفته شد.

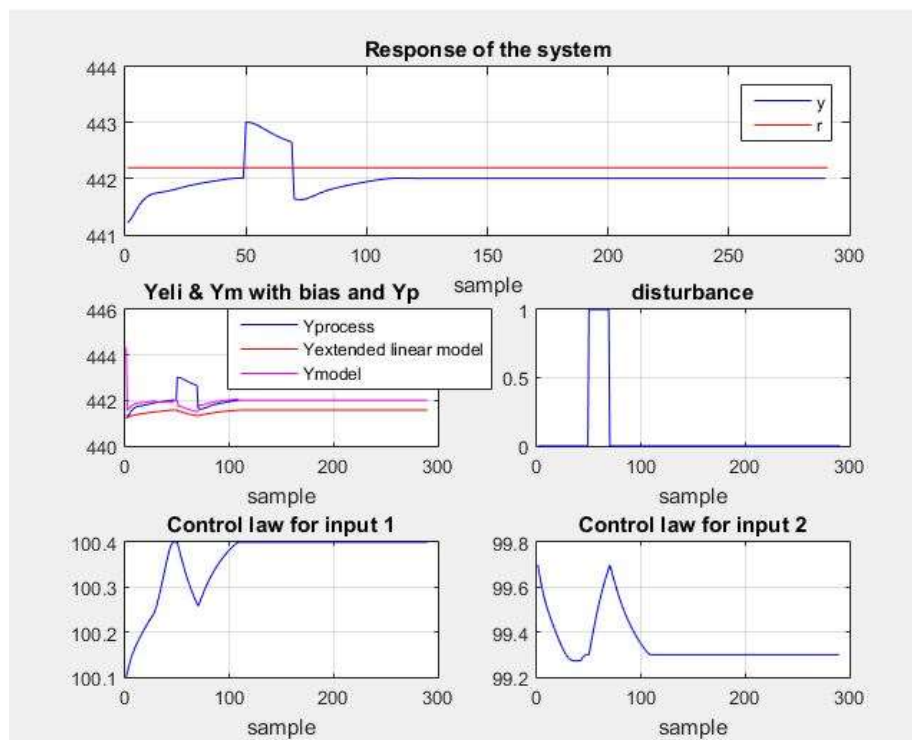


Figure 18. Responses of the system in verifying the effect of disturbance.

5.2. بررسی اثر نویز:

در این قسمت به خروجی پروسه نویز اضافه می‌کنیم. پاسخ سیستم در حضور نویز در شکل زیر آمده است.

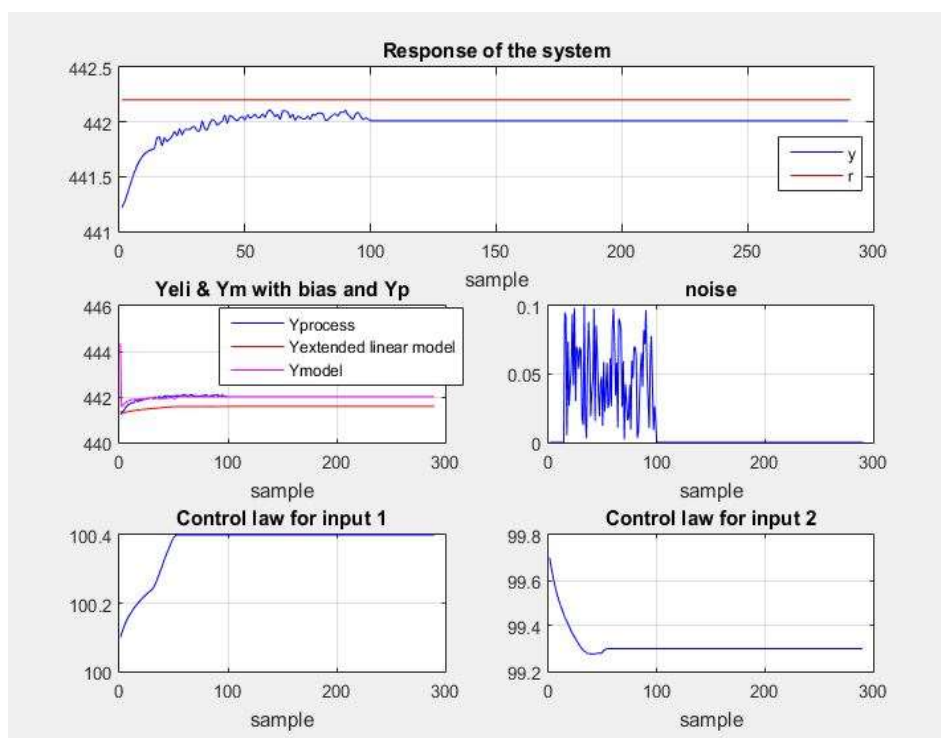


Figure 19. Responses of the system in verifying the effect of noise.

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 10% و 20% اضافه می کنیم. پاسخ سیستم به ازای این عدم قطعیت ها برای این پارامتر در شکل زیر آورده شده است.

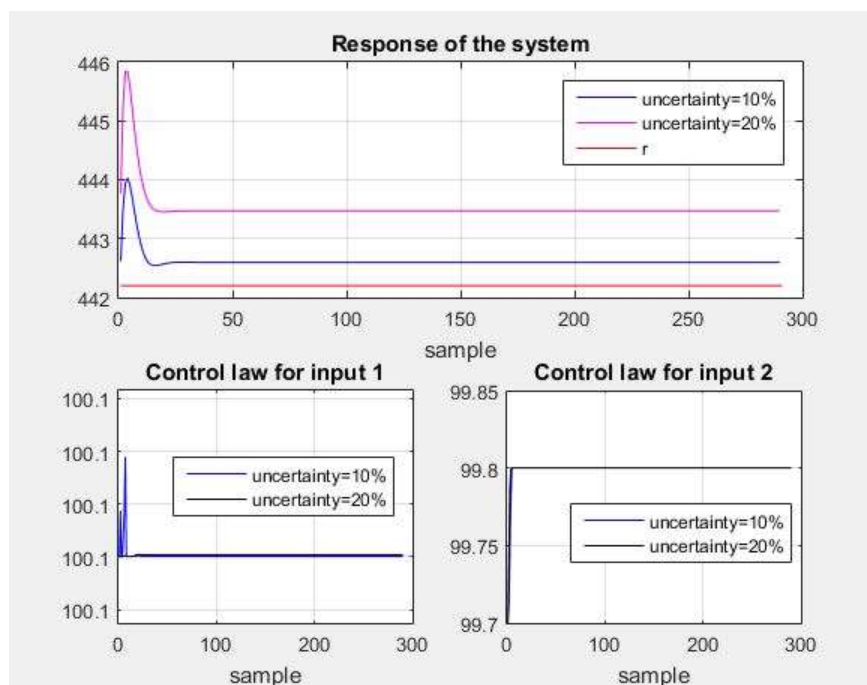


Figure 20. Responses of the system in verifying the effect of uncertainty on V .

همانطور که در شکل بالا می بینیم وجود عدم قطعیت باعث ایجاد خطای حالت ماندگار در پاسخ سیستم شده است و افزایش درصد عدم قطعیت میزان خطای حالت ماندگار سیستم را افزایش داده است.

اینبار عدم قطعیت با درصد های 10% و 20% را به پارامتر $CA0$ اضافه می کنیم. شکل زیر پاسخ سیستم در حضور عدم قطعیت های مذکور را نشان می دهد.

همانطور که در این شکل می بینیم عدم قطعیت روی این پارامتر نیز باعث ایجاد خطای حالت ماندگار شده است. می بینیم که در عدم قطعیت روی این پارامتر خطای حالت ماندگار بزرگتری ایجاد کرده است که این نشان دهنده این موضوع است که تاثیر پارامترهای مختلف در بروز وجود عدم قطعیت متفاوت است و بعضی پارامترها نسبت به عدم قطعیت حساس تر هستند.

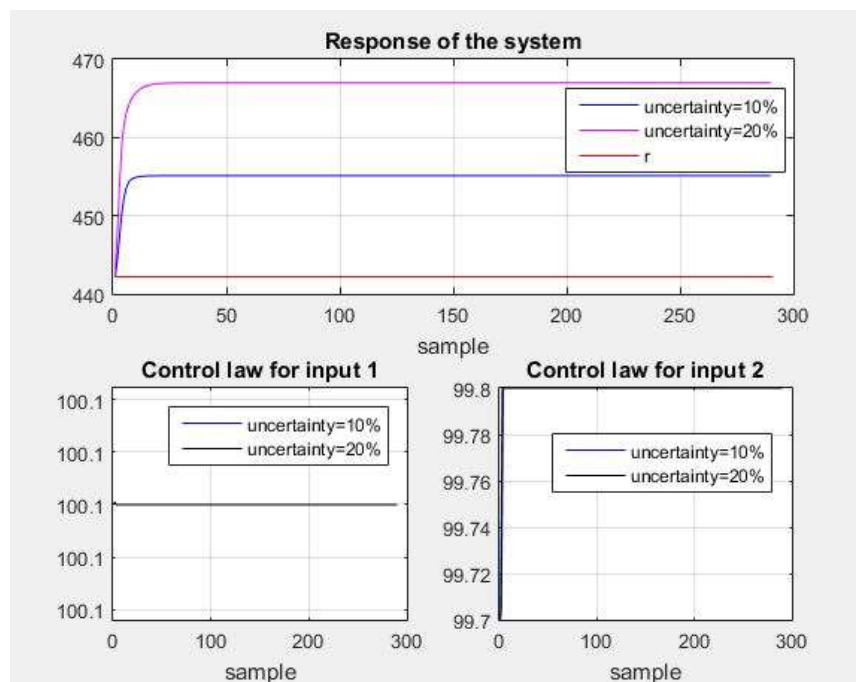


Figure 21. Responses of the system in verifying the effect of uncertainty on CA_0 .

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 مدل 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
q=100; V=100; Cas=.0792; dH=2e5; ro=1e3; Cp=1; roc=1e3; Cpc=1; qc=97; ha=7e5; Ts=443.5; 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==3
q=103; V=100; Cas=.0748; dH=2e5; ro=1e3; Cp=1; roc=1e3; Cpc=1; qc=97; ha=7e5; Ts=445.3; 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==4
q=97; V=100; Cas=0.1055; dH=2e5; ro=1e3; Cp=1; roc=1e3; Cpc=1; qc=103; ha=7e5; Ts=436.8; 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
end

```

کد زیر کدهای مربوط به طراحی کنترل کننده EDMC بدون محدودیت می باشد.

```
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;
P1=floor(tr1/Ts);
P2=floor(tr2/Ts);
P=max(P1,P2);
M=P-2;
%-----
gamma =1;
alpha=0.5;
betta=0.4;
delta=0.01;
% 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=((gain_DC2/gain_DC)^2)*gamma*gain_DC^2*eye(M);
% R2=gamma*gain_DC2^2*eye(M);
% R=[R1 zeros(M); zeros(M) R2];
%-----
x01=0.0882;
x02=441.2;
%-----
yre=[];
u_1=[];
u_2=[];
yeli=[];
ym=[];
y=0;

Y_d=zeros(P,length(t));
Y_past=zeros(P,length(t));
Y_m=zeros(P,length(t));
Y_eli=zeros(P,length(t));
Y_nm=zeros(P,length(t));
Y_per=zeros(P,length(t));

U1_ = zeros(P,length(t));
U2_ = zeros(P,length(t));
U_=[U1_; U2_];
```

```

U1per = zeros(P,length(t));
U2per = zeros(P,length(t));
Uper=[U1per; U2per];

U1nm = zeros(P,length(t));
U2nm = zeros(P,length(t));

dU1=zeros(M,length(t));
dU2=zeros(M,length(t));
dU=[dU1;dU2];

dext=zeros(1,length(t));
Dext=zeros(P,length(t));
Dnl=zeros(P,length(t));
E=zeros(P,length(t));

g=zeros(P,length(t));

g1=zeros(P,length(t));
g2=zeros(P,length(t));
Y_per1=zeros(P,length(t));
Y_per2=zeros(P,length(t));
Y_past1=zeros(P,length(t));
Y_past2=zeros(P,length(t));

U1=dU1;
U2=dU2;
% U1(1,1)=0.001;
% U2(1,1)=0.001;
n=zeros(1,length(t));

noise=zeros(length(t),1);
%noise(15:99,1)=0.01*rand(85,1);

%-----Step-----
r =ones(length(t),1);
%-----
%.....sine.....
%[r,t1]= gensig('sine',length(t)*Ts/2,length(t)*Ts,Ts);
%-----square wave-----
%r=0.03*441.2*[0*ones(floor(length(t)/5),1); 1*ones(floor(length(t)/5),1); 0*ones(floor(length(t)/5),1); -
1*ones(floor(length(t)/5),1); 0*ones(length(t)-4*floor(length(t)/5),1)];
%.....Step with various jump.....
%r=[1*ones(floor(length(t)/4),1); 3*ones(floor(length(t)/4),1); 5*ones(floor(length(t)/4),1); -1*ones(length(t)-
3*floor(length(t)/4),1)];

for i=1:length(t)-1

%-----
x01pa=x01;
x02pa=x02;
U1_(:,i+1)=(U1(1,i)+100)*ones(P,1);
U2_(:,i+1)=(U2(1,i)+100)*ones(P,1);

```

```

x011pa=x01;
x021pa=x02;
x012pa=x01;
x022pa=x02;
x011per=x01;
x021per=x02;
x012per=x01;
x022per=x02;
U1per(:,i+1)=(U1(1,i)+100)*(1+delta)*ones(P,1);
U2per(:,i+1)=(U2(1,i)+100)*(1+delta)*ones(P,1);
for j=1:P
    u1pa=U1_(j,i+1);
    u2pa=U2_(j,i+1);
    u11pa=U1_(j,i+1);
    u22pa=U2_(j,i+1);
    u11per=U1per(j,i+1);
    u22per=U2per(j,i+1);
    sim('Modelpa');
    x01pa=x1pa(end);
    x02pa=x2pa(end);
    Y_past(j,i+1)=x2pa(end)-441.2;
    x011pa=x1pa1(end);
    x021pa=x2pa1(end);
    Y_past1(j,i+1)=x2pa1(end);
    x012pa=x1pa2(end);
    x022pa=x2pa2(end);
    Y_past2(j,i+1)=x2pa2(end);
    x012per=x1per2(end);
    x022per=x2per2(end);
    Y_per2(j,i+1)=x2per2(end);
    x011per=x1per1(end);
    x021per=x2per1(end);
    Y_per1(j,i+1)=x2per1(end);
end

%-----

g1(1:P,i+1)=(Y_per1(1:P,i+1)-Y_past1(1:P,i+1))/(delta);
b1 = zeros(1,P); b1(1,1)= g1(1,i+1);
a1 = g1(1:P,i+1);
G1 = toeplitz(a1,b1);
G1(:,M) = G1(:,M:P)*ones(P-M+1,1);
G1 = G1(:,1:M);

g2(1:P,i+1)=(Y_per2(1:P,i+1)-Y_past2(1:P,i+1))/(delta);
b2 = zeros(1,P); b2(1,1)= g2(1,i+1);
a2 = g2(1:P,i+1);
G2 = toeplitz(a2,b2);
G2(:,M) = G2(:,M:P)*ones(P-M+1,1);
G2 = G2(:,1:M);
G=[G1 G2];
%-----

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

```

```

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

gain_DC = g1(end,i+1);
gain_DC2 = g2(end,i+1);
Q = eye(P);
R1 =((gain_DC2/gain_DC)^2)*gamma*gain_DC^2*eye(M);
R2=gamma*gain_DC2^2*eye(M);
R=[R1 zeros(M); zeros(M) R2];

K=(G'*Q*G+R)\(G'*Q);

x01nm=x01;
x02nm=x02;

%for k=1:20
while(1)

    Dext(:,i+1)=dext(i+1)*ones(P,1);
    E(:,i+1)=Y_d(:,i+1)-Y_past(:,i+1)-Dext(:,i+1)-Dnl(:,i+1);
    dU(:,i+1)=K*E(:,i+1);
    dU1(:,i+1)=dU(1:M,i+1);
    dU2(:,i+1)=dU(M+1:2*M,i+1);
    U1(:,i+1)=dU1(:,i+1)+U1(:,i);
    U2(:,i+1)=dU2(:,i+1)+U2(:,i);

    U1nm(1:M,i+1)=U1(:,i+1);
    U1nm(M+1:P,i+1)=U1(M,i+1);
    U2nm(1:M,i+1)=U2(:,i+1);
    U2nm(M+1:P,i+1)=U2(M,i+1);
    for j=1:P
        u1nm=U1nm(j,i+1)+100;
        u2nm=U2nm(j,i+1)+100;
        sim('Modelnm');
        x01nm=x1nm(end);
        x02nm=x2nm(end);
        Y_nm(j,i+1)=x2nm(end)-441.2;
    end

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

    Dnl(:,i+1)=(1-betta)*Dnl(:,i+1)+betta*(Y_nm(:,i+1)-Y_m(:,i+1));
    n(i+1)=n(i+1)+1;
    if (norm(Dnl(:,i+1)-old))/norm(old)<=0.02 || n(i+1)>=20
        break
    end

end

u1=U1(1,i+1)+100;

```

```

u2=U2(1,i+1)+100;
sim('Model');

yp=y(end)+noise(i,1);
dext(i+2)=yp-Y_nm(1,i+1)-441.2;
y=yp-441.2;
x01=x1(end);
x02=x2(end);
yre=[yre; yp];
yeli=[yeli; Y_eli(1,i+1)];
ym=[ym; Y_m(1,i+1)];
u_1=[u_1; u1];
u_2=[u_2; u2];

end

figure(1);
subplot(2,2,1);
plot(yre,'b');
hold on
plot(r+441.2,'r');
grid on
legend('y','r');
title('Response of the system');
xlabel('sample');
%-----
subplot(2,2,2);
plot(yre,'b');
hold on
plot(yeli+441.16,'r');
hold on
plot(ym+441.2,'m');
grid on
legend('y','r');
title('Yeli & Ym with bias and Yp');
legend('Yprocess','Yextended linear model','Ymodel');
xlabel('sample');
%-----
subplot(2,2,3);
plot(u_1,'b');
grid on
xlabel('sample');
title('Control law for input 1');
%-----
subplot(2,2,4);
plot(u_2,'b');
grid on
xlabel('sample');
title('Control law for input 2');

```

کد زیر مربوط به طراحی کنترل کننده EDMC با وجود محدودیت می باشد.

```

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;
P1=floor(tr1/Ts);
P2=floor(tr2/Ts);
P=max(P1,P2);
M=P-2;
%-----
gamma =1;
alpha=0.5;
betta=0.4;
delta=0.01;
% 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=((gain_DC2/gain_DC)^2)*gamma*gain_DC^2*eye(M);
% R2=gamma*gain_DC^2*eye(M);
% R=[R1 zeros(M); zeros(M) R2];
%-----
x01=0.0882;
x02=441.2;
%-----
up1=0.4;
up2=-0.2;
low1=0.1;
low2=-0.7;
% up1=0.4;
% up2=0.2;
% low1=-0.5;
% low2=-0.9;
yre=[];
u_1=[];
u_2=[];
x_1=[];
yeli=[];
ym=[];
y=0;

Y_d=zeros(P,length(t));
Y_past=zeros(P,length(t));
Y_m=zeros(P,length(t));
Y_eli=zeros(P,length(t));
Y_nm=zeros(P,length(t));
Y_per=zeros(P,length(t));

```

```

U1_ = zeros(P,length(t));
U2_ = zeros(P,length(t));
U_=[U1_; U2_];

```

```

U1per = zeros(P,length(t));
U2per = zeros(P,length(t));
Uper=[U1per; U2per];

```

```

U1nm = zeros(P,length(t));
U2nm = zeros(P,length(t));

```

```

dU1=zeros(M,length(t));
dU2=zeros(M,length(t));
dU=[dU1;dU2];

```

```

dext=zeros(1,length(t));
Dext=zeros(P,length(t));
Dnl=zeros(P,length(t));
E=zeros(P,length(t));

```

```

g=zeros(P,length(t));

```

```

g1=zeros(P,length(t));
g2=zeros(P,length(t));
Y_per1=zeros(P,length(t));
Y_per2=zeros(P,length(t));
Y_past1=zeros(P,length(t));
Y_past2=zeros(P,length(t));

```

```

U1=dU1;
U2=dU2;

```

```

dU01=dU;
dU01(1,1)=0.001;
dU01(M+1,1)=0.001;

```

```

n=zeros(1,length(t));

```

```

noise=zeros(length(t),1);
%noise(15:99,1)=0.1*rand(85,1);
dist=zeros(length(t),1);
%dist(50:69,1)=ones(20,1);

```

```

%-----Step-----
r =ones(length(t),1);
%-----
%.....sine.....
%[r,t1]= gensig('sine',length(t)*Ts/2,length(t)*Ts,Ts);
%-----square wave-----
%r=[0*ones(floor(length(t)/5),1);      1*ones(floor(length(t)/5),1);      0*ones(floor(length(t)/5),1);      -
1*ones(floor(length(t)/5),1); 0*ones(length(t)-4*floor(length(t)/5),1)];
%.....Step with various jump.....

```

```

%r=[1*ones(floor(length(t)/6),1);      3*ones(floor(length(t)/6),1);      5*ones(floor(length(t)/6),1);      -
1*ones(floor(length(t)/6),1); 0*ones(floor(length(t)/6),1); 4*ones(length(t)-5*floor(length(t)/6),1)];

```

```

for i=1:length(t)-1

```

```

%-----

```

```

x01pa=x01;
x02pa=x02;
U1_(:,i+1)=(U1(1,i)+100)*ones(P,1);
U2_(:,i+1)=(U2(1,i)+100)*ones(P,1);
x011pa=x01;
x021pa=x02;
x012pa=x01;
x022pa=x02;
x011per=x01;
x021per=x02;
x012per=x01;
x022per=x02;
U1per(:,i+1)=(U1(1,i)+100)*(1+delta)*ones(P,1);
U2per(:,i+1)=(U2(1,i)+100)*(1+delta)*ones(P,1);
for j=1:P
    u1pa=U1_(j,i+1);
    u2pa=U2_(j,i+1);
    u11pa=U1_(j,i+1);
    u22pa=U2_(j,i+1);
    u11per=U1per(j,i+1);
    u22per=U2per(j,i+1);
    sim('Modelpa');
    x01pa=x1pa(end);
    x02pa=x2pa(end);
    Y_past(j,i+1)=x2pa(end)-441.2;
    x011pa=x1pa1(end);
    x021pa=x2pa1(end);
    Y_past1(j,i+1)=x2pa1(end);
    x012pa=x1pa2(end);
    x022pa=x2pa2(end);
    Y_past2(j,i+1)=x2pa2(end);
    x012per=x1per2(end);
    x022per=x2per2(end);
    Y_per2(j,i+1)=x2per2(end);
    x011per=x1per1(end);
    x021per=x2per1(end);
    Y_per1(j,i+1)=x2per1(end);
end

```

```

end

```

```

%-----

```

```

g1(1:P,i+1)=(Y_per1(1:P,i+1)-Y_past1(1:P,i+1))/(delta);
b1 = zeros(1,P); b1(1,1)= g1(1,i+1);
a1 = g1(1:P,i+1);
G1 = toeplitz(a1,b1);
G1(:,M) = G1(:,M:P)*ones(P-M+1,1);
G1 = G1(:,1:M);

g2(1:P,i+1)=(Y_per2(1:P,i+1)-Y_past2(1:P,i+1))/(delta);

```

```

b2 = zeros(1,P); b2(1,1)= g2(1,i+1);
a2 = g2(1:P,i+1);
G2 = toeplitz(a2,b2);
G2(:,M) = G2(:,M:P)*ones(P-M+1,1);
G2 = G2(:,1:M);
G=[G1 G2];
%-----

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

gain_DC = g1(end,i+1);
gain_DC2 = g2(end,i+1);
Q = eye(P);
R1 =((gain_DC2/gain_DC)^2)*gamma*gain_DC^2*eye(M);
R2=gamma*gain_DC2^2*eye(M);
R=[R1 zeros(M); zeros(M) R2];

K=(G'*Q*G+R)\(G'*Q);

x01nm=x01;
x02nm=x02;

while(1)

    Dext(:,i+1)=dext(i+1)*ones(P,1);
    E(:,i+1)=Y_d(:,i+1)-Y_past(:,i+1)-Dext(:,i+1)-Dnl(:,i+1);
    H = 2*(G'*Q*G+R);
    H=(H+H')/2;
    f = -(2*E(:,i+1)'*Q*G)';
    T1 = ones(2*M);
    T = tril(T1);
    A = [T;-T];
    ub1=up1*ones(M,1)-U1(1,i);
    ub2=up2*ones(M,1)-U2(1,i);
    ub=[ub1; ub2];
    lb1=-low1*ones(M,1)+U1(1,i);
    lb2=-low2*ones(M,1)+U2(1,i);
    lb=[lb1; lb2];
    B=[ub; lb];
    opts = optimoptions('quadprog','Algorithm','interior-point-convex','Display','off');
    dU(:,i+1)= quadprog(H,f,A,B,[],[],[],dU01(:,i),opts);
    dU01(:,i+1)=dU(:,i+1);
    dU1(:,i+1)=dU(1:M,i+1);
    dU2(:,i+1)=dU(M+1:2*M,i+1);
    U1(:,i+1)=dU1(:,i+1)+U1(:,i);
    U2(:,i+1)=dU2(:,i+1)+U2(:,i);

    U1nm(1:M,i+1)=U1(:,i+1);
    U1nm(M+1:P,i+1)=U1(M,i+1);
    U2nm(1:M,i+1)=U2(:,i+1);
    U2nm(M+1:P,i+1)=U2(M,i+1);

```

```

for j=1:P
    u1nm=U1nm(j,i+1)+100;
    u2nm=U2nm(j,i+1)+100;
    sim('Modelnm');
    x01nm=x1nm(end);
    x02nm=x2nm(end);
    Y_nm(j,i+1)=x2nm(end)-441.2;
end

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

Dnl(:,i+1)=(1-betta)*Dnl(:,i+1)+betta*(Y_nm(:,i+1)-Y_m(:,i+1));
n(i+1)=n(i+1)+1;
if (norm(Dnl(:,i+1)-old))/norm(old)<=0.02 || n(i+1)>=20
    break
end

end

u1=U1(1,i+1)+100;
u2=U2(1,i+1)+100;
sim('Model');

yp=y(end)+noise(i,1)+dist(i,1);
dext(i+2)=yp-Y_nm(1,i+1)-441.2;
y=yp-441.2;
x01=x1(end);
x02=x2(end);
yre=[yre; yp];
yeli=[yeli; Y_eli(1,i+1)];
ym=[ym; Y_m(1,i+1)];
u_1=[u_1; u1];
u_2=[u_2; u2];
x_1=[x_1; x1(end)];

end

figure(1);
subplot(2,2,1:2);
plot(yre,'b');
hold on
plot(r+441.2,'r');
grid on
legend('y','r');
title('Response of the system');
xlabel('sample');
%-----
subplot(2,2,2);
plot(yre,'b');
hold on
plot(yeli+441.16,'r');
hold on

```

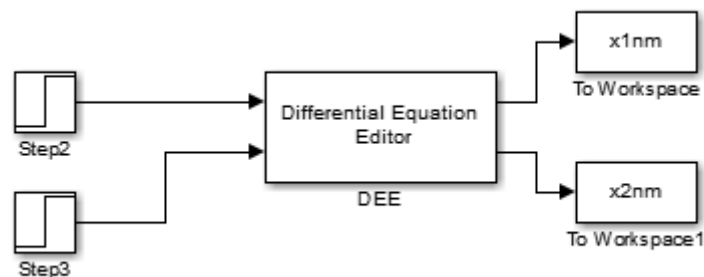
```

plot(ym+441.2,'m');
grid on
legend('y','r');
title('Yeli & Ym with bias and Yp');
legend('Yprocess','Yextended linear model','Ymodel');
xlabel('sample');
%-----
subplot(2,2,3);
plot(u_1,'b');
grid on
xlabel('sample');
title('Control law for input 1');
%-----
subplot(2,2,4);
plot(u_2,'b');
grid on
xlabel('sample');
title('Control law for input 2');
%-----
% subplot(3,2,5:6);
% plot(x_1)
% grid on
% xlabel('sample');
% title('state one');
% subplot(3,2,4);
% plot(dist,'b');
% grid on
% xlabel('sample');
% title('disturbance');
% plot(noise,'b');
% grid on
% xlabel('sample');
% title('noise');
% figure
% plot(x_1);

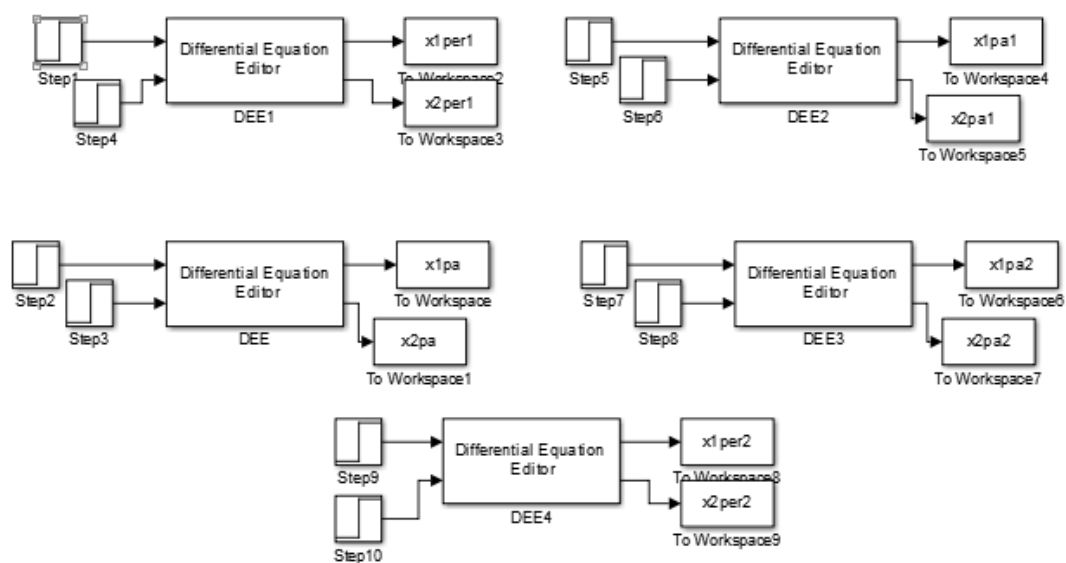
```

عکس شبیه سازی‌های انجام شده در سیمولینک متلب در شکل زیر آورده شده است.

شکل زیر مربوط به مدل غیر خطی می‌باشد.



شکل زیر مربوط به مدل‌هایی است که برای ساخت Y_{per} و Y_{past} مورد استفاده قرار گرفته است.



شکل زیر مربوط به پروسه می باشد.

