



5.2.1395

# پروژه اول درس MPC

پیاده سازی کنترلر MAC

نام دانشجو: فاطمه زاهدی

شماره دانشجویی: 94201583

نام استاد: دکتر حائری

## فهرست:

1. معرفی سیستم مورد استفاده: ..... 2
2. طراحی کنترل کننده MAC: ..... 2
  - 2.1. مک بر روی سیستم بعد از خطی سازی: ..... 4
  - 2.2. مک بر روی سیستم غیر خطی: ..... 5
3. پاسخ سیستم به ازای خروجی های مطلوب مختلف: ..... 6
  - 3.1. خروجی مطلوب پالس: ..... 6
  - 3.2. خروجی مطلوب سینوس: ..... 6
  - 3.3. خروجی مطلوب پله ای با پرش های مختلف: ..... 7
  - 3.4. خروجی مطلوب پله ای: ..... 8
4. بررسی پارامترها: ..... 9
  - 4.1. تغییر ماتریس R: ..... 9
  - 4.2. تغییر ماتریس Q: ..... 10
  - 4.3. تغییر مقدار N: ..... 11
  - 4.4. تغییر مقدار  $\alpha$ : ..... 11
  - 4.5. تغییر مقدار P و M: ..... 12
  - 4.6. تغییر مقدار  $T_s$ : ..... 13
5. بررسی اغتشاش و نویز: ..... 13
6. بررسی عدم قطعیت در پروسه: ..... 14
7. کدها و شبیه سازی ها: ..... 16

## 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.

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

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

با جایگذاری مقادیر پارامترهای مورد استفاده در این مدل و خطی سازی آن می توان کنترلر مک را برای این سیستم طراحی کرد.

## 2. طراحی کنترل کننده MAC:

در این طراحی ابتدا بایستی به تعیین پارامترهای  $N$ ,  $P$ ,  $T_s$  و  $M$  پردازیم. ابتدا مقدار  $T_s$  را مشخص می کنیم. به منظور مشخص کردن این پارامتر ابتدا پاسخ پله سیستم را به ازای هر کدام از ورودی ها می کشیم و فرض می کنیم که هر کدام از این ورودی ها بعد از 10 الی 20 سمپل به مقدار steady خود رسیده اند. بنابراین بازه ی مربوط به  $T_s$  به صورت زیر خواهد شد.

$$\frac{t_s}{20} \leq T_s \leq \frac{t_s}{10}$$

با توجه به رابطه بالا ما مقدار  $T_s$  را برابر با 0.1 گرفته ایم. در این طراحی مقدار  $M$  و  $P$  را برابر با هم قرار می دهیم. که این مقدار با توجه به رابطه  $M = P = \frac{t_r}{T_s}$  بدست 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 رابطه ی  $N = \frac{t_s}{T_s}$  بدست آوردیم و مقدار بزرگتر به عنوان  $N$  برای طراحی کنترلر استفاده شد که این مقدار برابر با 15 شد.

همانطور که گفته شد این مقادیر با استفاده از  $t_r$  و  $t_s$  که از پاسخ پله گسسته شده دو تابع تبدیل سیستم بدست می آید حاصل شد. پاسخ پله به ازای هر کدام از ورودی های سیستم در زیر آورده شده است.

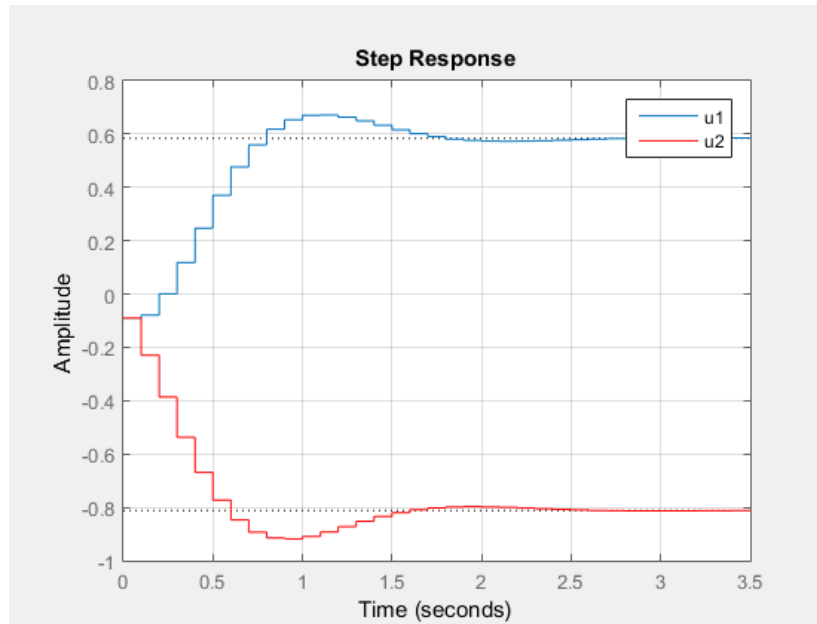


Figure 1. Step responses of the system.

به منظور طراحی کنترل کننده Mac بایستی ماتریس Toeplitz و Hankel را با استفاده از پاسخ ضربه سیستم به ازای هر کدام از ورودی ها تشکیل دهیم. پاسخ ضربه سیستم به ازای هر کدام از ورودی ها در زیر آورده شده است.

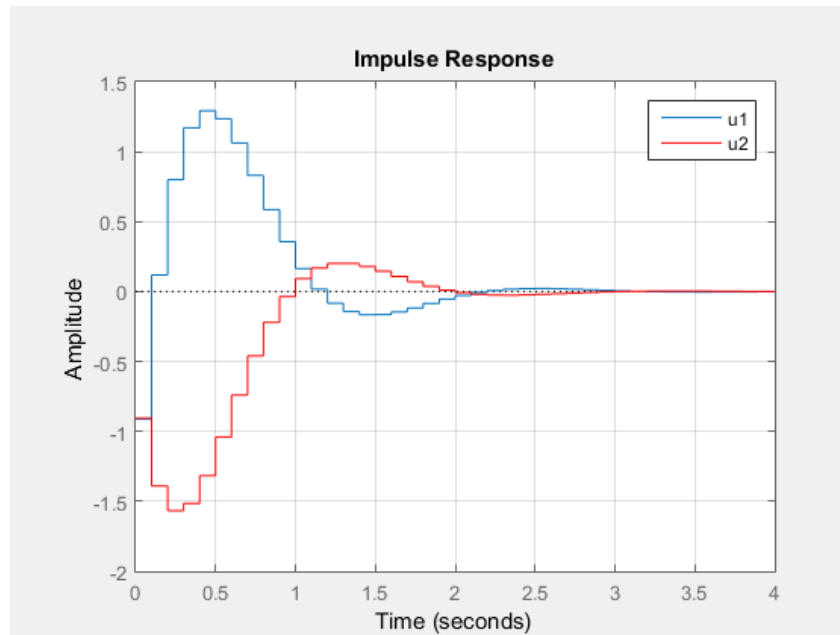


Figure 2. Impulse responses of the system.

با استفاده از این پاسخ ضربه ها ماتریس Toeplitz و Hankel را به ازای هر ورودی به صورت زیر تشکیل می دهیم و بعد ماتریس ها را کنار هم قرار داده و ماتریس Toeplitz و Hankel اصلی را می سازیم.

$$H = \begin{pmatrix} h_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ h_p & \cdots & h_1 \end{pmatrix}, \quad H^- = \begin{pmatrix} h_2 & \cdots & h_N \\ \vdots & \ddots & \vdots \\ h_{p+1} & \cdots & 0 \end{pmatrix}, \quad H = [H_1 \ H_2], \quad H^- = [H_1^- \ H_2^-]$$

بردار  $U$  مربوط به ورودی های آینده سیستم و بردار  $U^-$  مربوط به ورودی های گذشته سیستم می باشد. این دو بردار در زیر آورده شده اند.

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

$Y_m$  که خروجی مدل است به صورتی که در آورده شده است می باشد.

$$Y_m = HU + Y_{past}, \quad Y_{past} = H^- U^-$$

هدف بدست آوردن ورودی های آینده  $U$  به گونه ای است که تابع هدف زیر مینیمم گردد و خروجی تا حد امکان خروجی مطلوب را دنبال نماید.

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

در اینجا  $Y_p$  خروجی پروسه که از سیستم غیر خطی بدست می آید و  $Y_d$  خروجی مطلوب می باشد.

با توجه به اینکه سیستم را در حالت حلقه بسته بررسی می کنیم  $D$  مخالف صفر است و از رابطه زیر بدست می آید.

$$Y_p = Y_m + D$$

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 می دهیم. از طرفی چون دو ورودی داریم به ازای هر ورودی  $R_1$  و  $R_2$  را بدست می آوریم و روی قطر ماتریس  $R$  قرار می دهیم. ماتریس  $R$  به صورت زیر بدست می آید.

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

که این  $r$  به منظور هم وزن کردن  $R$  های ورودی ها با هم است زیرا به خاطر گینشان ممکن است تفاوت وزن ایجاد شود که در این سیستم ما این  $r$  را  $(1.4)^2$  گرفته ایم.

با استفاده از روابط بالا کنترل کننده مک را به صورت programmed طراحی کرده و این کنترل کننده را یک بار بر روی سیستم خطی سازی شده و یک بار بر روی سیستم غیر خطی اجرا می کنیم

## 2.1. مک بر روی سیستم بعد از خطی سازی:

ابتدا به منظور اینکه از درستی طراحی خود مطمئن شویم  $Y_p$  را همان خروجی سیستممان بعد از خطی سازی، می گذاریم و نتایج به صورت زیر شد. شکل زیر مربوط به آلفا 0.5 می باشد.

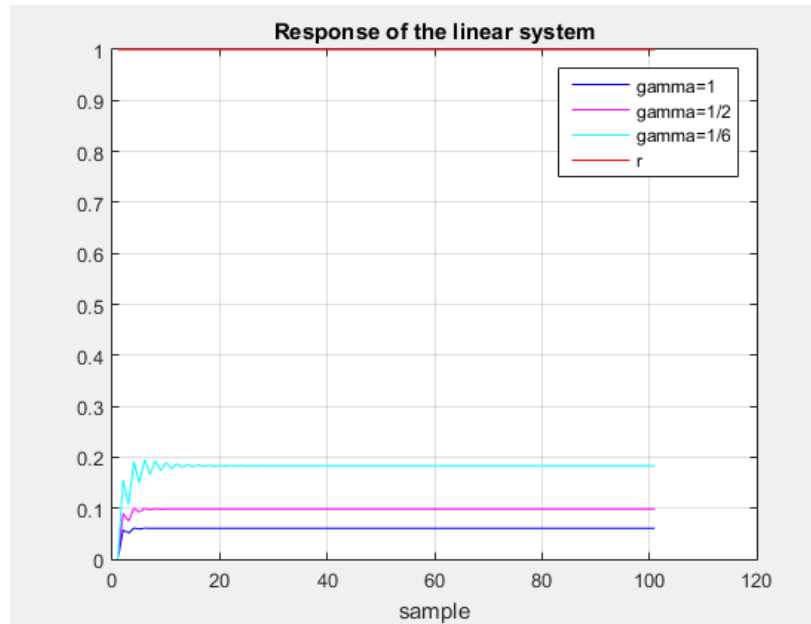


Figure 3. MAC on linearization system.

در شکل بالا به وضوح می‌توان تاثیر کاهش گاما در کم شدن بایاس و در نتیجه کم شدن خطا در steady state را دید.

## 2.2. مک بر روی سیستم غیر خطی:

این بار  $Y_P$  خروجی است که از سیستم غیر خطی گرفته شده است و نتایج را با وجود کنترل کننده مک برای این خروجی می‌بینیم. در زیر به ازای گامای 0.5 و آلفا 0.5 خروجی سیستم رسم شده است.

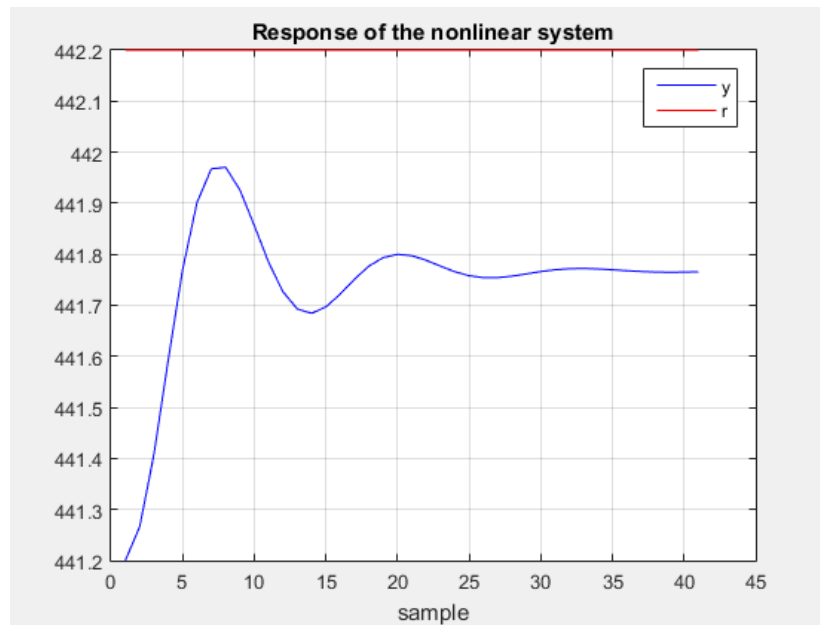


Figure 4. MAC on nonlinear system.

همانطور که مشاهده می شود بین خروجی مطلوب و خروجی سیستم مقداری بایاس وجود دارد که این موضوع با توجه به اینکه گاما را 0.5 انتخاب کرده ایم طبیعی است. بررسی اثر پارامتر گاما را در بخش های بعد خواهیم دید. لازم به ذکر است که نقطه کار خروجی سیستم برابر با 441.2 می باشد.

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

در کلیه قسمت های این بخش گاما را 0.5 و آلفا را 0.3 قرار داده ایم.

#### 3.1. خروجی مطلوب پالس:

در این قسمت به ازای خروجی مطلوب سیستم پالس می دهیم و پاسخ سیستم به صورت زیر در آمد.

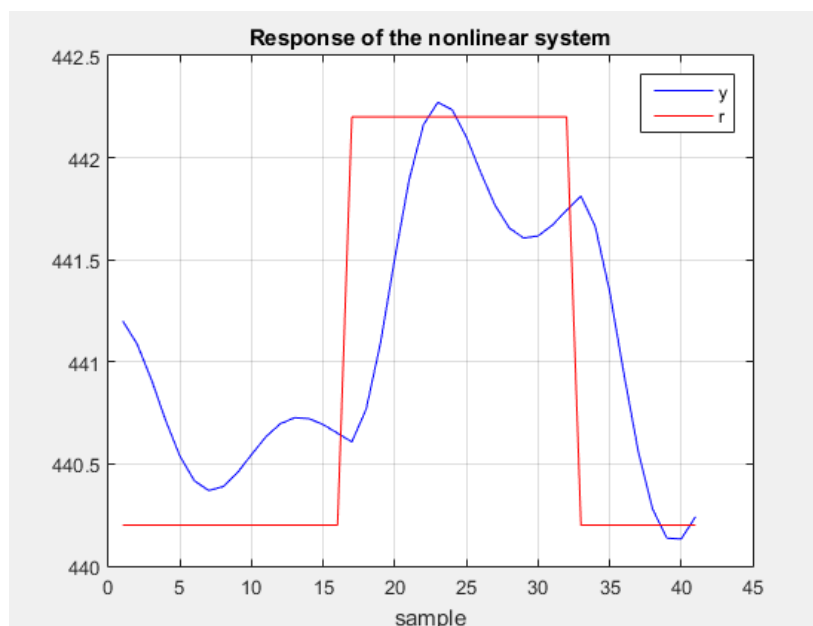


Figure 5. Response of the system with pulse as a desired output.

#### 3.2. خروجی مطلوب سینوس:

در این قسمت به ازای خروجی مطلوب سیستم سیگنال سینوسی می دهیم و پاسخ سیستم به ازای این خروجی مطلوب به صورت زیر شد. همانطور که در این شکل می بینیم خروجی سیستم به صورت سینوسی در آمد که تقریباً دارد خروجی مطلوب را ترک می کند.

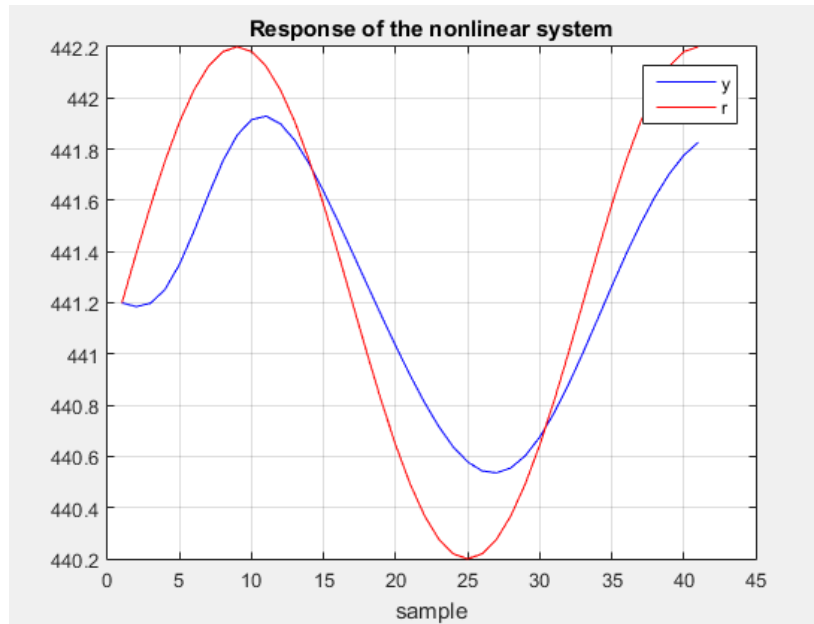


Figure 6. Response of the system with sine as a desired output.

### 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 می توان دید پاسخ به ازای این خروجی مطلوب خیلی مناسب نیست و خروجی مطلوب را ترک کرده است.

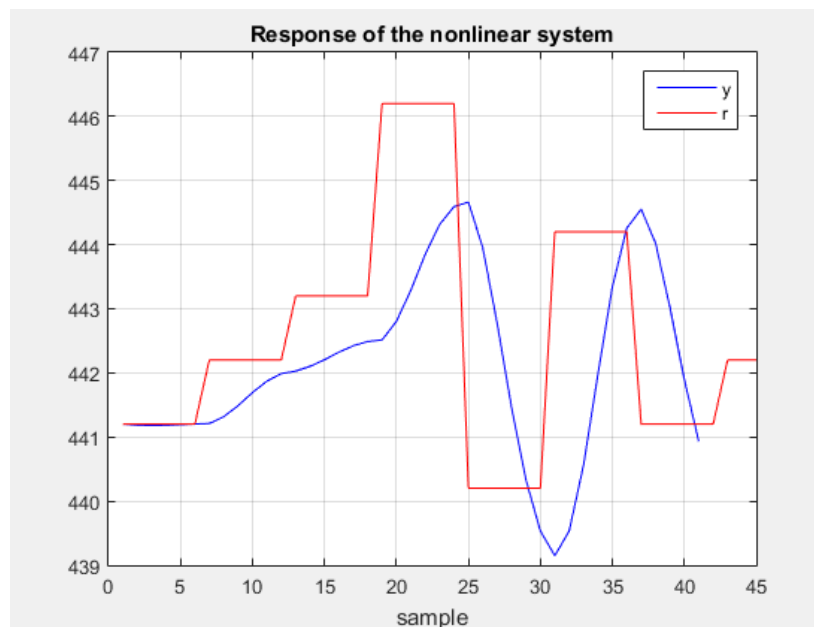


Figure 7. Response of the system with step with various jumps as a desired output.



### 3.4. خروجی مطلوب پله ای:

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 1%، 1.5% و 5% نقطه کار رسم می کنیم. شکل زیر پاسخ را به ازای این خروجی مطلوب با دامنه 1% نقطه کار نشان می دهد.

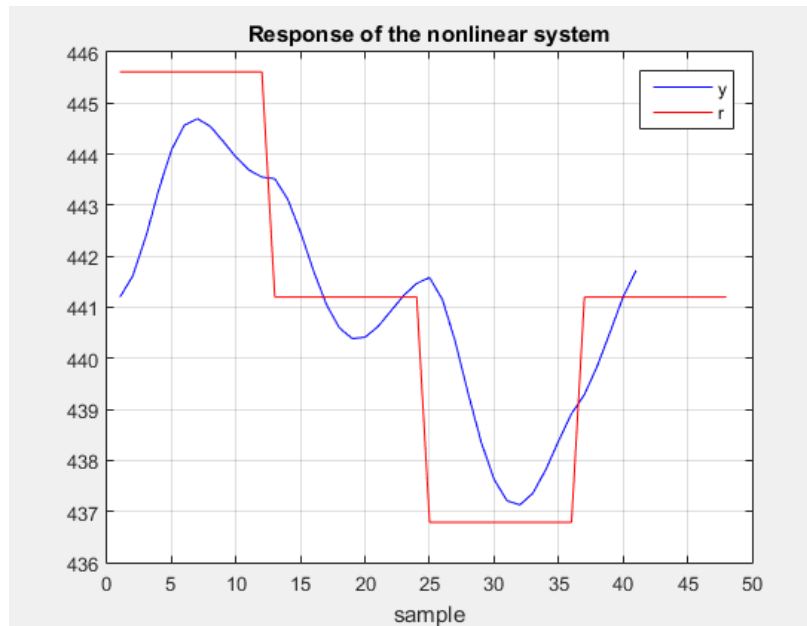


Figure 8. Response of the system with step with 1% of operating point as a desired output.

در زیر پاسخ سیستم به ازای همین خروجی مطلوب اما با دامنه 1.5% نقطه کار را می بینیم.

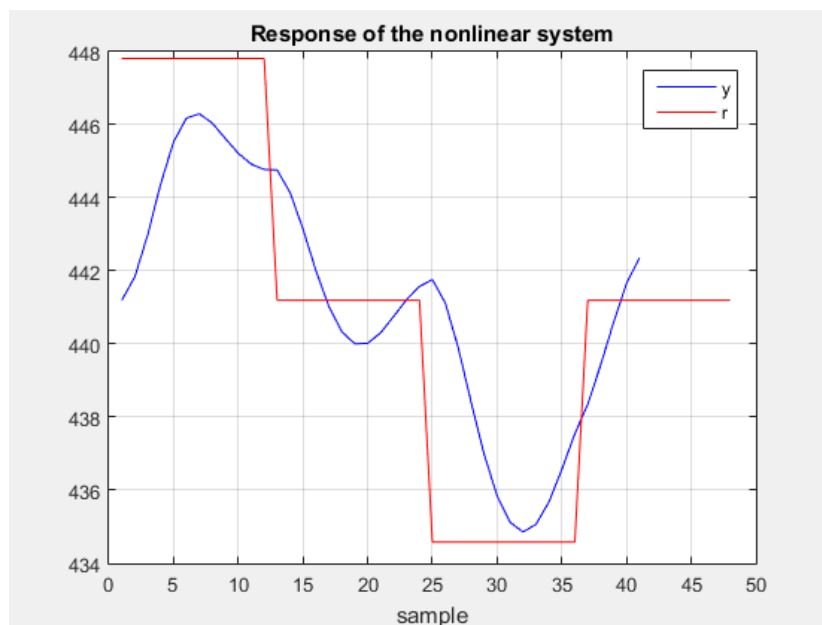


Figure 9. Response of the system with step with 1.5% of operating point as a desired output.

در پایان پاسخ سیستم به ازای همین خروجی مطلوب اما با دامنه 5% نقطه کار را می بینیم.

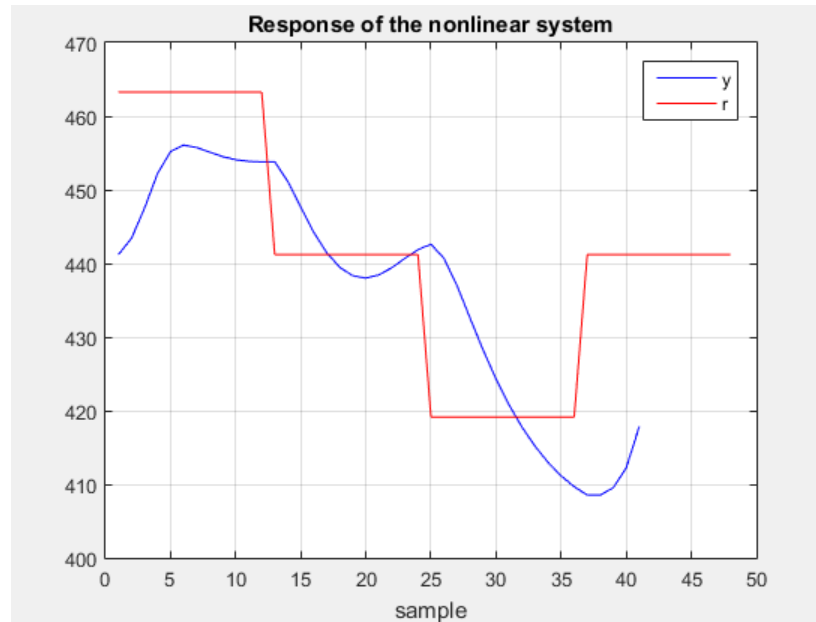


Figure 10. Response of the system with step with 3% of operating point as a desired output.

با توجه به سه شکل بالا می بینیم که هر چه درصد دامنه خروجی مطلوب را نسبت به نقطه کار افزایش می دهیم، پاسخ سیستم از خروجی مطلوب دورتر می شود تا جایی که می توان دید که به ازای دامنه 5% نقطه کار، خروجی دارد کم کم با خروجی مطلوب اختلاف زیادی پیدا می کند.

## 4. بررسی پارامترها:

در این بخش تاثیر تغییرات پارامترهای مورد استفاده در طراحی کنترلر مک 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 1/6 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 می بینیم با کاهش گاما از مقدار یک به 1/2 مقدار بایاس کاهش یافت و همچنین با کاهش گاما از 1/2 به 1/6 نیز بایاس کاهش یافت. بنابراین می توان در این شکل تاثیر کاهش گاما را در کاهش بایاس و در نتیجه کاهش خطای حالت ماندگار سیستم دید.

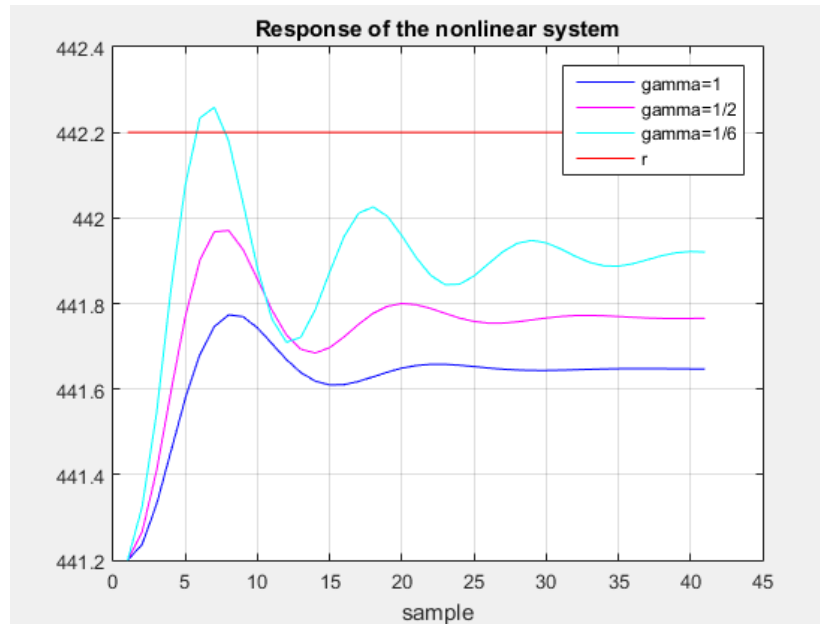


Figure 11. Responses of the system in verifying  $R$ .

## 4.2. تغییر ماتریس $Q$ :

در این قسمت به ازای گامای 1 و آلفای 0.5 پاسخ سیستم را به ازای  $Q=I$  و  $Q=2I$  بررسی می کنیم. انتظار می رود که این حالت پاسخ سیستم نیز مشابه حالتی که گاما را از 1 به 0.5 تغییر می دادیم شود.

شکل زیر پاسخ سیستم به ازای  $Q$  های مذکور را نشان می دهد.

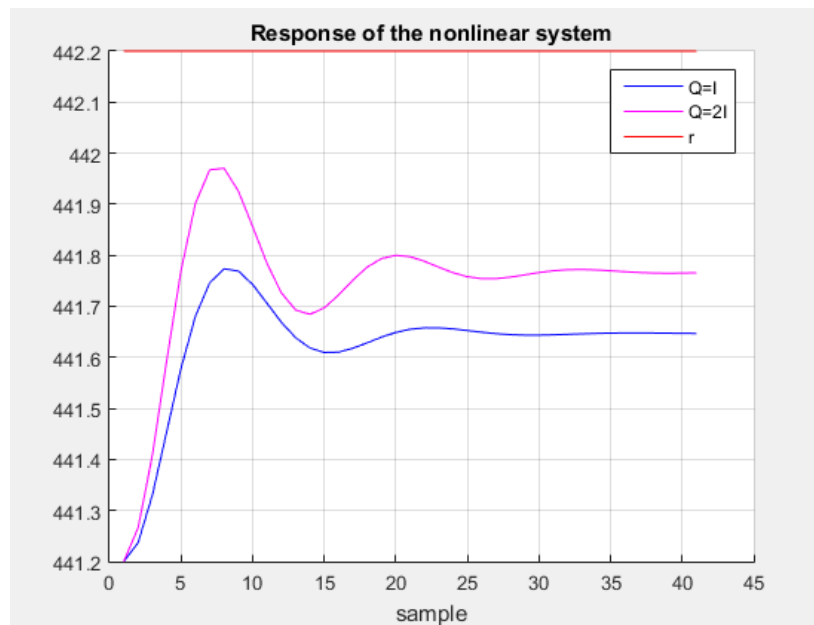


Figure 12. Responses of the system in verifying  $Q$ .

همانطور که می بینیم با افزایش  $Q$  خطای پاسخ کاهش یافته است و  $performance$  پاسخ افزایش پیدا کرده است.

### 4.3. تغییر مقدار $N$ :

در این قسمت پاسخ سیستم را به ازای سه  $N$  مختلف شامل 10، 15 و 30 بررسی می کنیم. هر دو مقدار گاما و آلفا برابر 0.5 در طی سه بررسی قرار داده شده است. شکل زیر پاسخ سیستم به ازای این سه  $N$  نشان می دهد. همانطور که مشاهده می کنید تغییر  $N$  تاثیری روی پاسخ سیستم نگذاشته است و هر سه پاسخ بر روی هم افتاده اند.

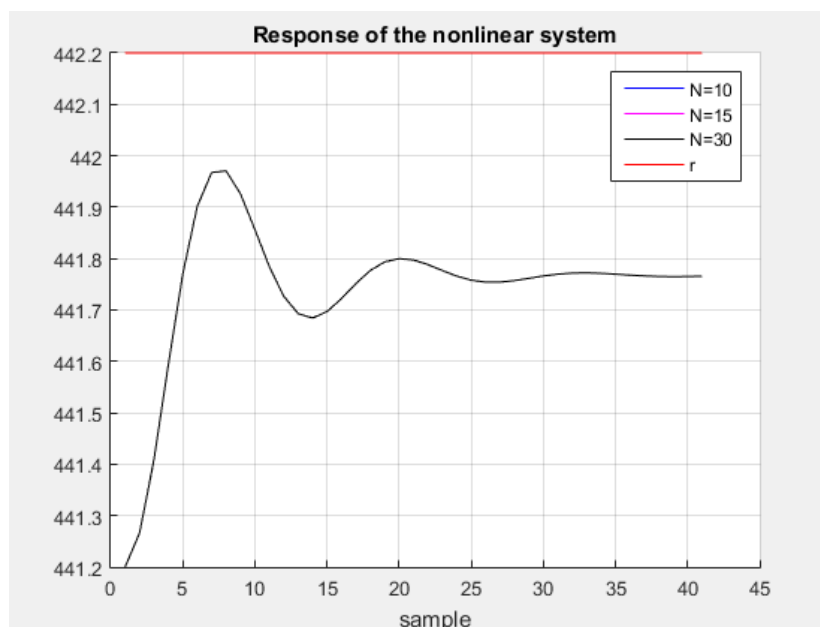


Figure 13. Responses of the system in verifying  $N$ .

### 4.4. تغییر مقدار $\alpha$ :

در این قسمت تاثیر پارامتر آلفا را بر روی پاسخ سیستم بررسی می کنیم. شکل زیر پاسخ سیستم به ازای مقادیر آلفا 0.3، 0.5 و 0.8 نشان می دهد. در طی این بررسی مقدار گاما 0.5 قرار داده شده است.

همانطور که در این شکل می بینیم با کاهش مقدار آلفا پاسخ سیستم تند تر شده و به تبع تند شدن پاسخ سیستم مقدار بایاس نیز کاهش یافته است. بنابراین می توان ادعا کرد که کاهش آلفا پاسخ سیستم را تندتر می کند.

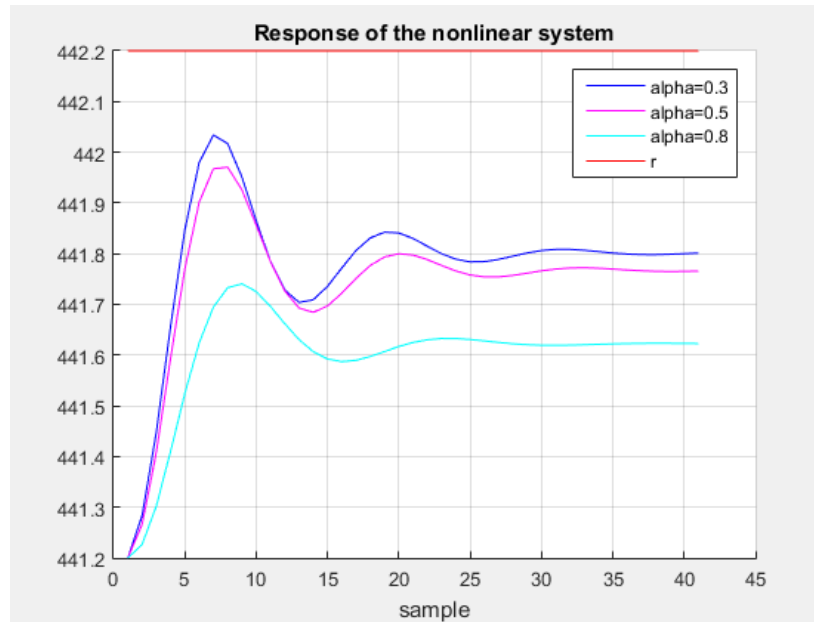


Figure 14. Responses of the system in verifying  $\alpha$ .

#### 4.5. تغییر مقدار P و M:

در این بخش مقدار P و M را که هر دو برابر هستند تغییر می دهیم و پاسخ سیستم را به ازای P های 3، 5 و 10 رسم می کنیم. شکل زیر پاسخ سیستم به ازای P های مختلف نشان می دهد. مقدار گاما و آلفا مقدار 0.5 گذاشته شده اند.

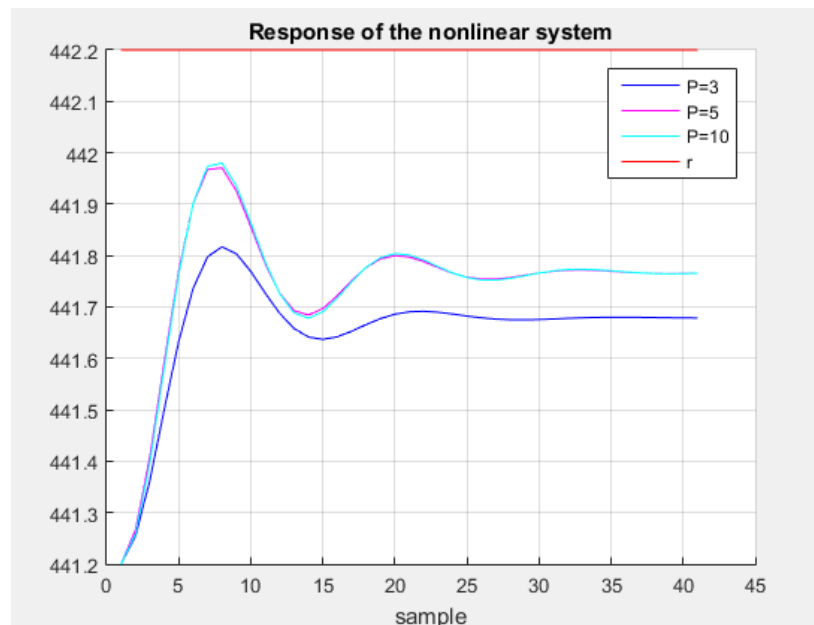


Figure 15. Responses of the system in verifying P or M.

همانطور که در این شکل می بینیم با افزایش P از مقدار 3 به 5 خطای حالت ماندگار (بایاس) کاهش یافته است و پاسخ سیستم تند تر شده است. اما با افزایش P از مقدار 5 که با استفاده از رابطه  $t_r/T_s$  بدست آمده بود به مقدار 10 پاسخ تغییری نکرد پس

به نظر می‌رسد که این افزایش  $P$  تا یک جایی روی بهبود پاسخ سیستم تاثیر می‌گذارد و از یک مقدار به بعد بهبودی حاصل نمی‌کند.

#### 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 0.25 نشان می‌دهد. همانطور که در این شکل می‌بینیم با کاهش مقدار  $T_s$  پاسخ سیستم بهتر شده است و خطای حالت ماندگار سیستم کاهش یافته است و سیستم تندتر شده است اما این بهبود پاسخ سیستم تا مقداری از  $T_s$  تاثیر می‌گذارد و از آن پس با کاهش دادن زمان نمونه برداری تغییری در پاسخ سیستم ایجاد نشده است.

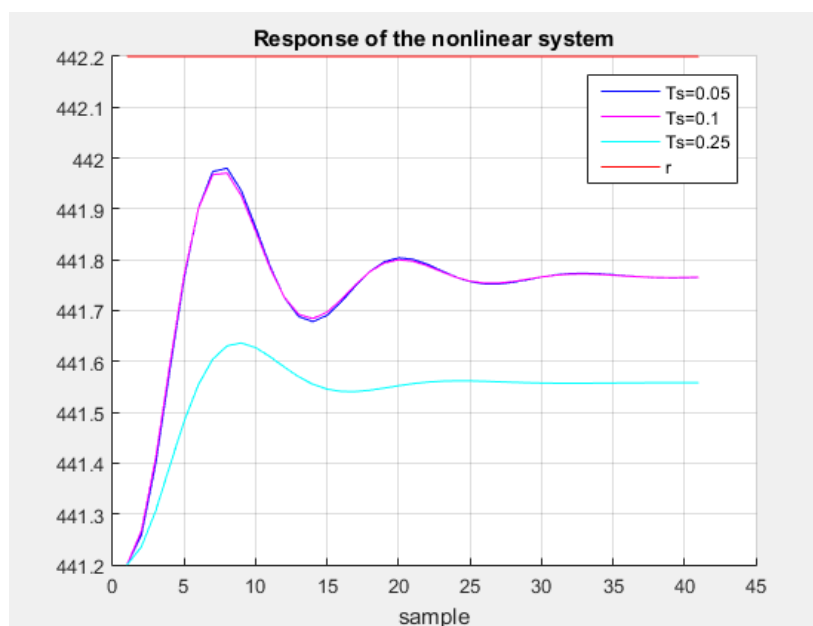


Figure 16. Responses of the system in verifying  $T_s$ .

#### 5. بررسی اغتشاش و نویز:

در این بخش روی خروجی پروسه نویز سفید با power ای برابر با 0.01 اضافه می‌کنیم و اثر این نویز را بر روی پاسخ سیستم مشاهده می‌کنیم.

با توجه به اینکه کد ها به صورتی نوشته شده است که سیستم غیر خطی از یک شبیه سازی که در سیمولینک انجام شده فراخوانی می‌شود. این نویز سفید را ما در همین محیط سیمولینک به خروجی پروسه اضافه کرده ایم. کد ها و عکس شبیه سازی های انجام شده در پیوست این گزارش آورده شده است.

شکل زیر پاسخ سیستم با وجود اضافه شدن نویز سفید به خروجی پروسه را نشان می دهد. لازم به ذکر است که مقدار گاما و آلفا از این به بعد برابر با 0.5 در نظر گرفته شده است.

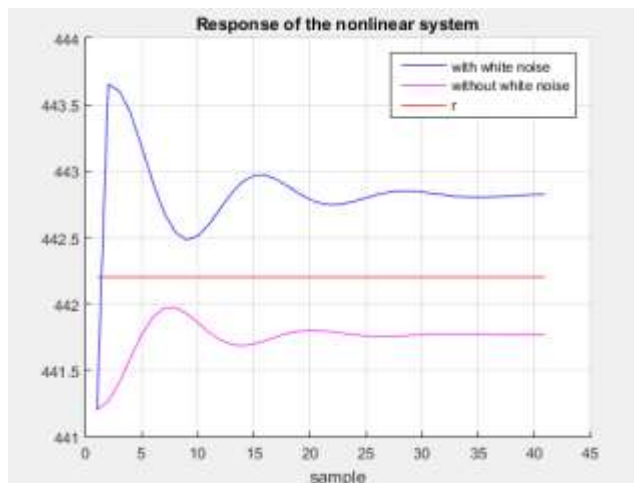


Figure 17. Responses of the system in verifying the effect of noise.

همانطور که در این شکل می بینیم نویز باعث شده است که خروجی سیستم در ابتدا (حالت گذرا) یک spike بزند ولی در حالت ماندگار کنترل کننده این رفتار را جبران کرده و بایاس در حالت ماندگار تقریباً با حالتی که نویز نداریم یکسان است و فقط کمی نسبت به حالت بدون نویز خطای حالت ماندگار کم 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%، 10%، 15% اضافه می کنیم. پاسخ سیستم به ازای این عدم قطعیت ها برای این پارامتر در شکل زیر آورده شده است.

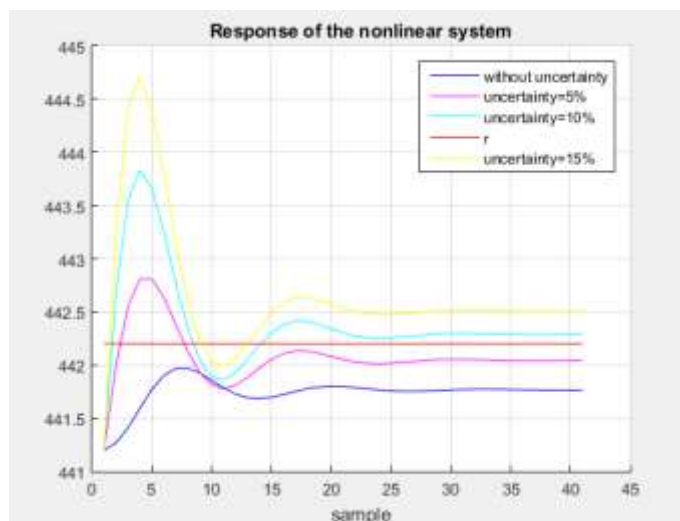


Figure 18. Responses of the system in verifying the effect of uncertainty on  $V$ .

همانطور که در شکل بالا می توان دید اضافه شدن عدم قطعیت بیشتر حالت گذرای پاسخ را خراب و دچار نوسان با pick بزرگ می کند و در پاسخ حالت ماندگار سیستم تاثیر بدی نمی گذارد. اما می توان این موضوع را دید که با افزایش درصد عدم قطعیت میزان بایاس نیز نسبت به درصد های پایین تر افزایش می یابد.

اینبار عدم قطعیت با درصد های 5%، 1% و 0.1% را به پارامتر  $C_{A0}$  اضافه می کنیم. شکل زیر پاسخ سیستم در حضور عدم قطعیت های مذکور را نشان می دهد.

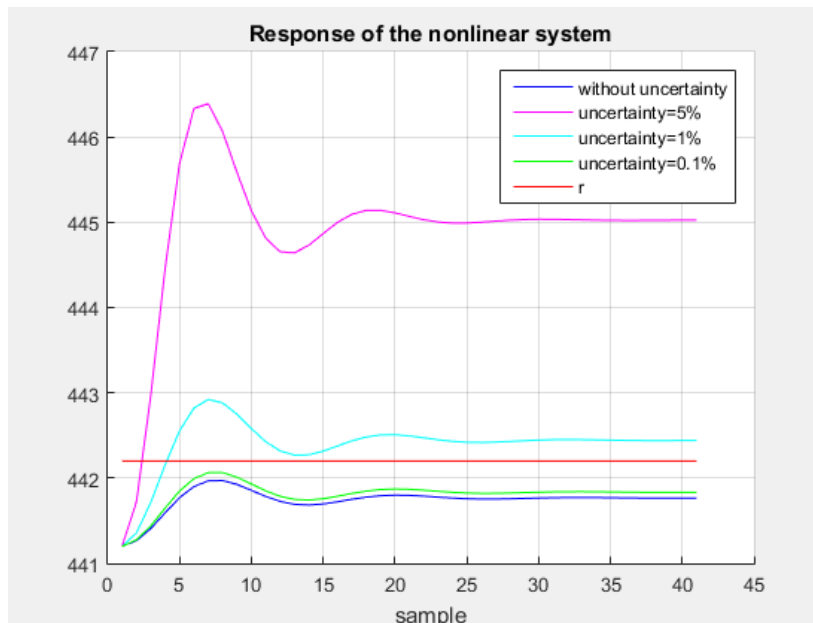


Figure 19. Responses of the system in verifying the effect of uncertainty on  $C_{A0}$ .

همانطور که در این شکل می بینیم عدم قطعیت با درصد 0.1% پاسخ سیستم را تغییر نداده است. اما وقتی درصد عدم قطعیت را افزایش داده ایم خطای حالت ماندگار زیاد شده است و این افزایش خطای ماندگار با افزایش درصد عدم قطعیت بیشتر نیز شده است. از طرفی عدم قطعیت باعث افزایش pick در حالت گذرا نیز شده است.



## 7. کدها و شبیه سازی ها:

کد زیر تابعی است که سیستم خطی سازی شده در آن قرار گرفته است به منظور اینکه در کد اصلی برای طراحی کنترل کننده مک سیستم را با استفاده از این کد فراخوانی کنیم. علاوه بر سیستم اصلی به منظور چک کردن درستی طراحی کنترل کننده دو سیستم دیگر در این تابع قرار داده شده که کنترلر بر روی آنها با فراخوانی آنها نیز چک 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
```

کد زیر کنترل کننده مک را در حالی که پروسه سیستم خطی سازی شده می باشد چک می کند. در این کد پروسه به صورت کد داخل خود برنامه نوشته شده است.

```
close all
clear
clc
[n1,d1,n2,d2]=Inputsys(1);
Gs1 = tf(n1,d1);
Gd1 = c2d(Gs1,0.25,'impulse');
[num1,den1]=tfdata(Gd1,'v');
Gs2 = tf(n2,d2);
Gd2 = c2d(Gs2,0.25,'impulse');
[num2,den2]=tfdata(Gd2,'v');
figure(1);
step(Gd1);
hold on
step(Gd2,'r')
grid on
sys_info = stepinfo(Gd1);
ts = sys_info.SettlingTime;
tr=sys_info.RiseTime;
Ts = 0.25;
N = ts/Ts;
t = 0:0.25:30;
[h1,t1] =lsim(Gd1,[1 zeros(1,120)],t);
[h2,t2] =lsim(Gd2,[1 zeros(1,120)],t);
figure(2)
impulse(Gd1,t)
P = tr/Ts; M = P; % In this method Control Horizon is equal to Predictive Horizon
% .....Toeplitz Matrix.....
b = zeros(1,P);
b(1,1)= h1(2);
a = h1(2:P+1);
H1 = toeplitz(a,b);
H1(:,M)=H1(:,M:P)*ones(P-M+1,1);
H1=H1(:,1:M);
b2 = zeros(1,P);
b2(1,1)= h2(2);
a2 = h2(2:P+1);
H2 = toeplitz(a2,b2);
H2(:,M)=H2(:,M:P)*ones(P-M+1,1);
H2=H2(:,1:M);
H=[H1 H2];
% .....Hankel Matrix.....
c = h1(3:P+2);
r = [(h1(P+2:N+1))' zeros(1,P-1)];
H_1 = hankel(c,r);
c2 = h2(3:P+2);
r2 = [(h2(P+2:N+1))' zeros(1,P-1)];
H_2 = hankel(c2,r2);
H_=[H_1 H_2];
% .....
gamma =1/10;
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 = gamma*gain_DC^2*eye(M);
R2=((2.5)^2)*gamma*gain_DC^2*eye(M);
R=[R1 zeros(M); zeros(M) R2];
Kmac = inv(H'*Q*H+R)*H'*Q;
y(1) = 0;
alpha = 0.5;
U_ = zeros(2*N-2,1);
U=zeros(2*M,1);
%r = ones(150+P+1,1);
[r,t1]= gensig('sine',20,40,0.25);
ypast = zeros(150+P,1);
d(1)= 0;
ym=zeros(150+P,1);
y_1 = 0;
y_2 = 0;
x1(1)=0; x2(1)=0;

for t = 1:150
y(t+1)=num1(1)*U(1) +num1(2)*U_(2) -den1(2)*y_1+num2(1)*U(M+1)+num2(2)*U_(N+1)-den1(3)*y_2;
y_2 = y_1;
y_1 = y(t+1);
yd(t+1)=y(t+1);

for i = 1:P
yd(t+i+1) = alpha*yd(t+i)+(1-alpha)*r(t+i+1); % Programmed
%yd(t+i) = alpha*yd(t+i-1)+(1-alpha)*r(t);% Unprogrammed
end
%.....
d(t+1) = y(t+1) - ym(1);
E = (yd(t+2:t+P+1))' - ypast(t+1:t+P) - d(t+1)*ones(P,1);
U = Kmac * E;
ym = H*U + H_*U_;
%.....
U_(2:N-1) = U_(1:N-2);
U_(1) = U(1);
U_(N+1:2*N-2) = U_(N:2*N-3);
U_(N) = U(M+1);
u(t) = U(1);
ypast(t+2:t+P+1) = H_*U_;
end
%
figure(3)
plot(r)
hold on
plot(y,'r');

```

کد زیر با استفاده از تابع مذکور سیستم خطی را فراخوانی می کند و طراحی کنترلر مک را انجام می دهد. در این کد پروسه از سیمولینک فراخوانی می شود و طوری نوشته شده که در همین کد هم می توان پروسه غیر خطی را فراخوانی کرد و نتیجه کنترلر را روی پاسخ آن دید هم می توان پروسه خطی را فراخوانی کرد که پروسه خطی به منظور چک کردن درستی طراحی کنترلر آورده شده است. در این کد نیز تغییر پارامتر ها و تغییر خروجی مطلوب نیز آورده شده است فقط لازم به ذکر است که به صورت comment آورده شده اند و هر قسمت را بایستی uncomment کنیم تا نتیجه را برای آن قسمت ببینیم.

```

%close all
clear
clc
[n1,d1,n2,d2]=Inputsys(1);
Gs1 = tf(n1,d1);
Gd1 = c2d(Gs1,0.1,'impulse');
[num1,den1]=tfdata(Gd1,'v');
Gs2 = tf(n2,d2);
Gd2 = c2d(Gs2,0.1,'impulse');
[num2,den2]=tfdata(Gd2,'v');
% figure(1)
% step(Gd1)
% hold on
% step(Gd2,'r');
% grid on
% legend('u1','u2');
sys_info = stepinfo(Gd1);
ts = sys_info.SettlingTime;
tr=sys_info.RiseTime;
sys_info = stepinfo(Gd2);
ts2 = sys_info.SettlingTime;
tr2=sys_info.RiseTime;
%Ts=0.05;
Ts = 0.1;
%Ts=0.25;
N=floor( ts/Ts);
%N=30;
%N=10
t = 0:0.1:4;
[h1,t1] =lsim(Gd1,[1 zeros(1,40)],t);
[h2,t2] =lsim(Gd2,[1 zeros(1,40)],t);
% figure(2)
% impulse(Gd1,t);
% hold on
% impulse(Gd2,t,'r');
% grid on
% legend('u1','u2');
%P=3;
P =floor(tr/Ts);
%P=10;
M = P; % In this method Control Horizon is equal to Predictive Horizon
% .....Toeplitz Matrix.....
b = zeros(1,P);
b(1,1)= h1(2);
a = h1(2:P+1);
H1 = toeplitz(a,b);
H1(:,M)=H1(:,M:P)*ones(P-M+1,1);
H1=H1(:,1:M);
b2 = zeros(1,P);
b2(1,1)= h2(2);
a2 = h2(2:P+1);
H2 = toeplitz(a2,b2);
H2(:,M)=H2(:,M:P)*ones(P-M+1,1);
H2=H2(:,1:M);
H=[H1 H2];
% .....Hankel Matrix.....

```

```

c = h1(3:P+2);
r = [(h1(P+2:N+1))' zeros(1,P-1)];
H_1 = hankel(c,r);
c2 = h2(3:P+2);
r2 = [(h2(P+2:N+1))' zeros(1,P-1)];
H_2 = hankel(c2,r2);
H_=[H_1 H_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 =((1.4)^2)*gamma*gain_DC^2*eye(M);
R2=gamma*gain_DC2^2*eye(M);
R=[R1 zeros(M); zeros(M) R2];
Kmac = (H'*Q*H+R)\(H'*Q);
%alpha=0.3;
alpha=0.5;
%alpha=0.8;
x01=0.0882;
x02=441.2;

U1_ = zeros(N-1,length(t));
U2_ = zeros(N-1,length(t));
U_=[U1_ ; U2_];
d=zeros(1,length(t));
%y1=0; %linear
y1=441.2;
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_bar=zeros(P,length(t));
U1=zeros(M,length(t));
U2=zeros(M,length(t));
% .....step.....
r =ones(length(t),1);
% .....pulse.....
% [r,t1]= gensig('square',8,10,0.25);
% r=r;
% for l = 1:length(r)
% if (r(l)==0)
% r(l) = -1;
% end
% end
% .....sine.....
%[r,t1]= gensig('sine',8,10,0.25);
% .....step with various jump.....
% r1=[0 0 0 0 0 1 1 1 1 1 2 2 2 2 2 5 5 5 5 5 -1 -1 -1 -1 -1 3 3 3 3 3];
% r=[r1; r1];
% .....Step.....
% r1=5*4.412*[1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 0 0 0 0 0 0 0 0 0 0 0];
% r=[r1];
% .....

```

```

U=[U1;U2];

for i=1:length(t)-1

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

Y_past(:,i)=H_*U(:,i);
D(:,i)=d(i)*ones(P,1);

E_bar(:,i)=Y_d(:,i)-Y_past(:,i)-D(:,i);
U(:,i)=Kmac*E_bar(:,i);
U1(:,i)=U(1:M,i);
U2(:,i)=U(M+1:2*M,i);

U(:,i)=[U1(:,i);U2(:,i)];

Y_m(:,i)=H*U(:,i)+Y_past(:,i);

U1_(2:N-1,i+1) = U1_(1:N-2,i+1);
U1_(1,i+1)=U1(1,i);
U2_(2:N-1,i+1) = U2_(1:N-2,i+1);
U2_(1,i+1)=U2(1,i);

U_(:,i+1)=[U1_(:,i+1);U2_(:,i+1)];

u1=U1(1,i);
u2=U2(1,i);
sim('Model')
%d(i+1)=yl(end)-Y_m(1,i); %linear
d(i+1)=y(end)-Y_m(1,i);
%y1=[y1;yl(end)]; % linear
y1=[y1; y(end)+441.2];
x01=x1(end);
x02=x2(end);
%y=yl(end); % linear
y=y(end); % nonlinear
end
figure(3);
plot(y1,'b');
hold on
plot(r+441.2,'r');
grid on
%axis([0 45 439 447]);
legend('y','r');
title('Response of the nonlinear system');
xlabel('sample');
% ..... Linear .....
% plot(y1,'b');
% hold on
% % plot(r,'r');

```

```
% grid on
% title('Response of the linear system');
% xlabel('sample');
```

عکس شبیه سازی انجام شده در سیمولینک در زیر آورده شده است. که همانطور که در این شکل می بینید هنگامی که میخواهیم نویز به خروجی پروسه اضافه کنیم بلوک نویز با بلوک جمع کننده متصل می کنیم.

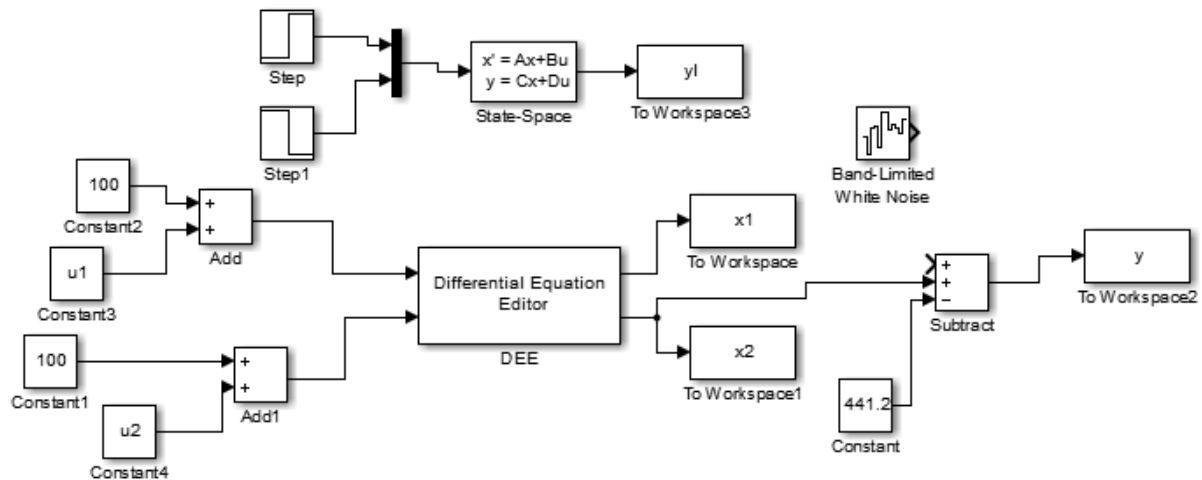


Figure 20. Simulation of the linear and nonlinear plant in Simulink.