



03.1395

پروژه پنجم درس MPC

طراحی کنترل کننده MMPC

نام دانشجو: فاطمه زاهدی

شماره دانشجویی: 94201583

نام استاد: دکتر حائری

دانشکده مهندسی برق

فهرست:

1. معرفی سیستم مورد استفاده: 2
2. طراحی کنترل کننده MMPC: 3
 - 2.1 MMPC بدون محدودیت: 3
 - 2.2 MMPC با محدودیت: 4
3. پاسخ سیستم به ازای خروجی مطلوب‌های مختلف: 5
 - 3.1 خروجی مطلوب سینوس: 5
 - 3.2 خروجی مطلوب پله‌ای با پرش‌های مختلف: 7
 - 3.3 خروجی مطلوب موج مویی: 8
4. بررسی اثر پارامترها: 10
 - 4.1 بررسی پارامتر گاما (ماتریس R): 10
 - 4.2 بررسی تغییر ماتریس Q : 13
 - 4.3 بررسی پارامتر α : 14
 - 4.4 بررسی پارامتر P : 15
 - 4.5 بررسی پارامتر M : 16
 - 4.6 بررسی پارامتر T_s : 17
5. بررسی اثر اغتشاش و نویز: 18
 - 5.1 بررسی اثر اغتشاش: 18
 - 5.2 بررسی اثر نویز: 19
6. بررسی عدم قطعیت در پروسه: 20
7. کدها و شبیه سازی ها: 21

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.

این سیستم در سه نقطه کار متفاوت خطی سازی شده است و روابط آن در زیر آورده شده است. هر کدام از این مدل ها در طراحی کنترل کننده MMPC استفاده شده است که ترتیب مدل های مورد استفاده به ترتیب همین نقاط کاریست که در زیر آورده شده است.

نقطه کار اول:

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

معادلات فضای حالت سیستم پس از خطی سازی حول این نقطه کار و جایگذاری پارامترها:

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

نقطه کار دوم:

$$C_A=0.0792, \quad T=443.5, \quad q=100, \quad q_c=97.$$

معادلات فضای حالت سیستم پس از خطی سازی حول این نقطه کار و جایگذاری پارامترها:

$$A = \begin{pmatrix} -12.6 & 0 \\ 2322.4 & 7.4 \end{pmatrix}, \quad B = \begin{pmatrix} 0.0092 & 0 \\ -0.935 & -0.9294 \end{pmatrix}, \quad C = (0 \quad 1), \quad D = (0 \quad 0).$$

نقطه کار سوم:

$$C_A=0.0748, \quad T=445.3, \quad q=103, \quad q_c=97.$$

معادلات فضای حالت سیستم پس از خطی سازی حول این نقطه کار و جایگذاری پارامترها:

$$A = \begin{pmatrix} -13.7 & 0 \\ 2544 & 7.6 \end{pmatrix}, \quad B = \begin{pmatrix} 0.0093 & 0 \\ -0.953 & -0.9472 \end{pmatrix}, \quad C = (0 \quad 1), \quad D = (0 \quad 0).$$

2. طراحی کنترل کننده MMPC:

در این طراحی بایستی سه کنترل کننده gpc با استفاده از سه مدلی که در بخش قبل آورده شده است طراحی می‌کنیم. این طراحی مطابق با روشی که در طراحی gpc گفته شده بود انجام می‌گیرد. پس از اینکه هر کنترل کننده قانون کنترلی خود را تولید کرد این du ها با وزن‌هایی با هم جمع می‌شوند و قانون کنترلی سیستم غیر خطی را می‌سازند. لازم به ذکر است که هر کدام از این مدل‌ها نیز از طریق همین قانون کنترلی که از میانگین وزنی قانون‌های کنترلی بدست آمده کنترل می‌شوند. بدین معنی که خروجی‌های مدل نیز از طریق این قانون کنترلی ساخته می‌شوند. با توجه به خواسته مساله که بایستی در هر لحظه تنها دو مدل روشن شوند و وزن مدل سوم صفر باشد، برای رسیدن به این خواسته بعد از اینکه از طریق روابط زیر وزن هر کدام از مدل‌ها انتخاب شد دو مدلی که وزن بیشتری دارند (با توجه به روابط وزن‌ها خطای کمتری دارند) برای ساخت قانون کنترلی انتخاب می‌شوند و پس از آن دوباره نرمالایز می‌شوند.

$$w_i = \frac{1/e_i}{\sum_{j=1}^3 1/e_j}$$

2.1 MMPC بدون محدودیت:

بعد از پیاده سازی این کنترل کننده پاسخ سیستم غیر خطی و قانون‌های کنترلی به ازای حالتی که خروجی مطلوب پله با دامنه 445.3 به سیستم دادیم به صورت زیر در آمد.

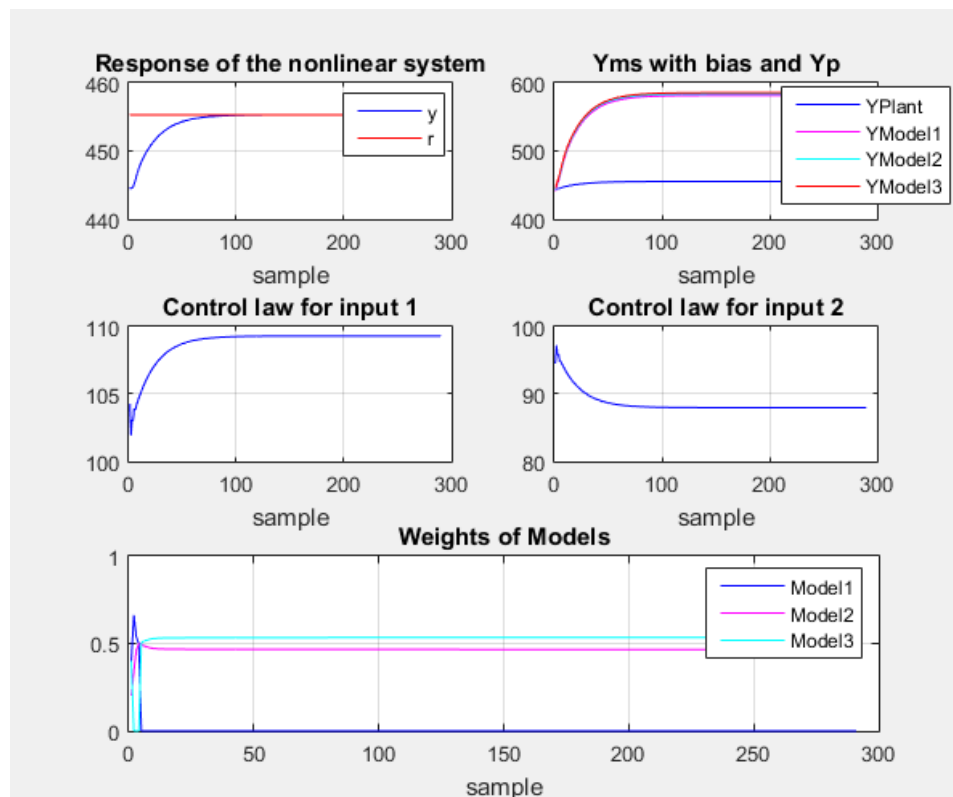


Figure1. MMPC on nonlinear system without constraint.

همانطور که در شکل بالا می‌بینیم از یه جایی به بعد مدل 3 و 2 انتخاب شده‌اند که این موضوع بخاطر نزدیک بودن دامنه پله واحد به مدل 3 و بعد از آن به مدل 2 کاملاً طبیعی و قابل انتظار است.

2.2 MMPC با 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.

$$95 \leq u_1 \leq 104, \quad 85 \leq u_2 \leq 95, \quad 0.05 \leq x_1 \leq 0.068, \quad 445.7 \leq x_2 \leq 454.14, \quad y = x_2.$$

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 ها 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. پس از محاسبه محدودیت‌ها بر حسب u و اشتراک گیری بین محدوده‌ها، محدوده‌های بدست آمده برای ورودی‌ها به صورت زیر شد.

$$95 \leq u_1 \leq 103, \quad 85 \leq u_2 \leq 90.$$

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.

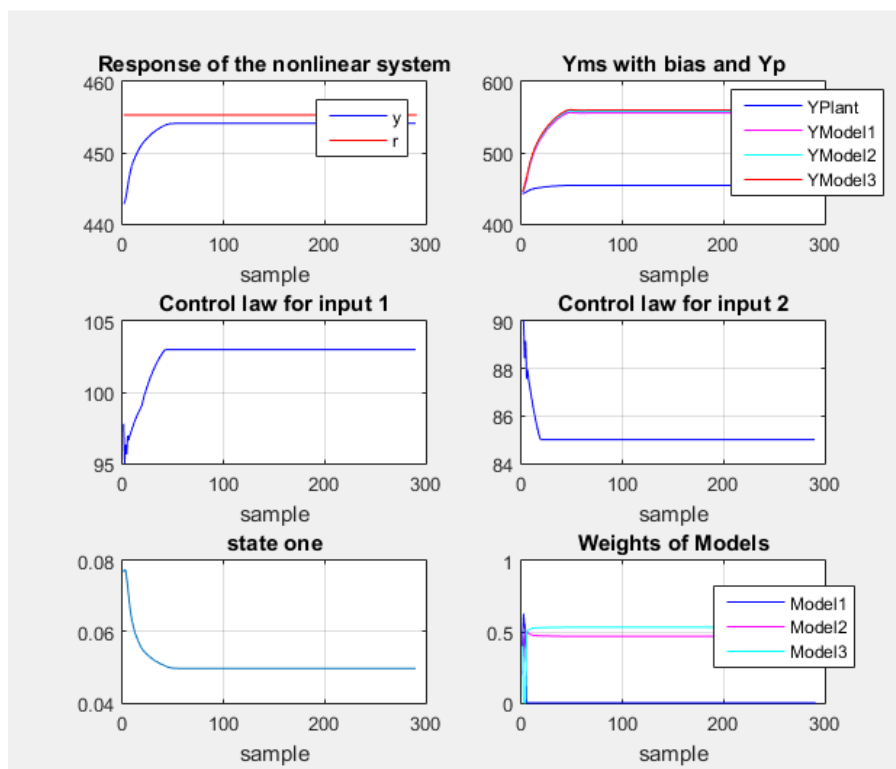


Figure2. MMPC on nonlinear system with constraint.

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

به منظور اینکه تاثیر سایر حد بالاها و حد پایین‌هایی که در این شکل خودشان را بخاطر نوع خروجی مطلوب نشان نداده‌اند ما پاسخ سیستم به ازای خروجی مطلوب سینوسی را نیز رسم کرده‌ایم که در شکل زیر آورده شده است.

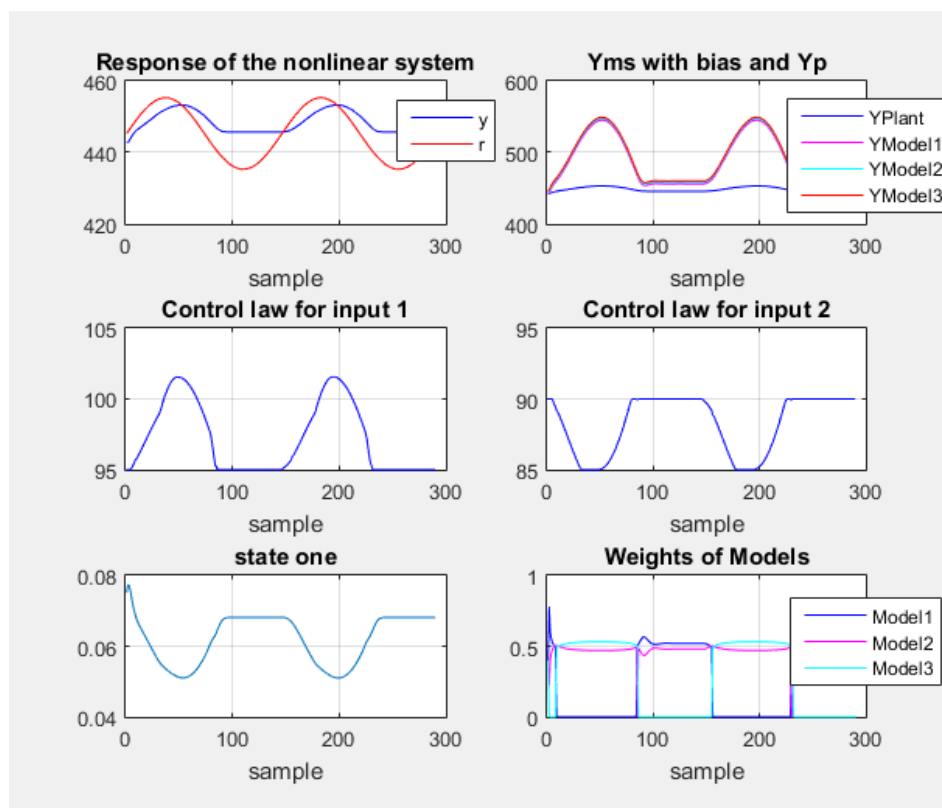


Figure3. MMPC on nonlinear system with constraint with sin wave as set point.

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

3. پاسخ سیستم به ازای خروجی مطلوب‌های مختلف:

در این بخش خروجی‌های مطلوب مختلفی را به سیستم می‌دهیم و پاسخ سیستم را مشاهده می‌کنیم.

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 وضوح دید انتخاب می‌کنیم و در آن به بررسی می‌پردازیم. با توجه به اینکه ما سه مدل و در نتیجه سه نقطه کار داریم در نتیجه پاسخ سیستم به ازای موج سینوسی حول سه نقطه کار را بررسی می‌کنیم.

پاسخ سیستم به موج سینوسی حول نقطه کار مدل اول در شکل زیر آورده شده است.

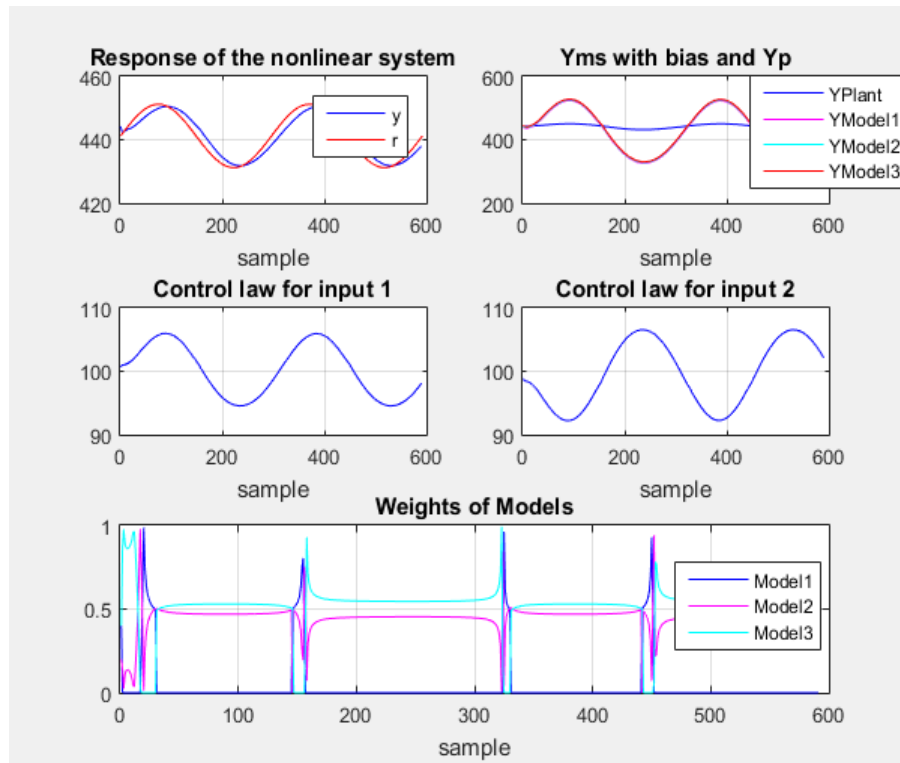


Figure 4. Response of the system with sine wave around operation point of model 1 as a desired output.

پاسخ سیستم به موج سینوسی حول نقطه کار مدل دوم در شکل زیر آورده شده است.

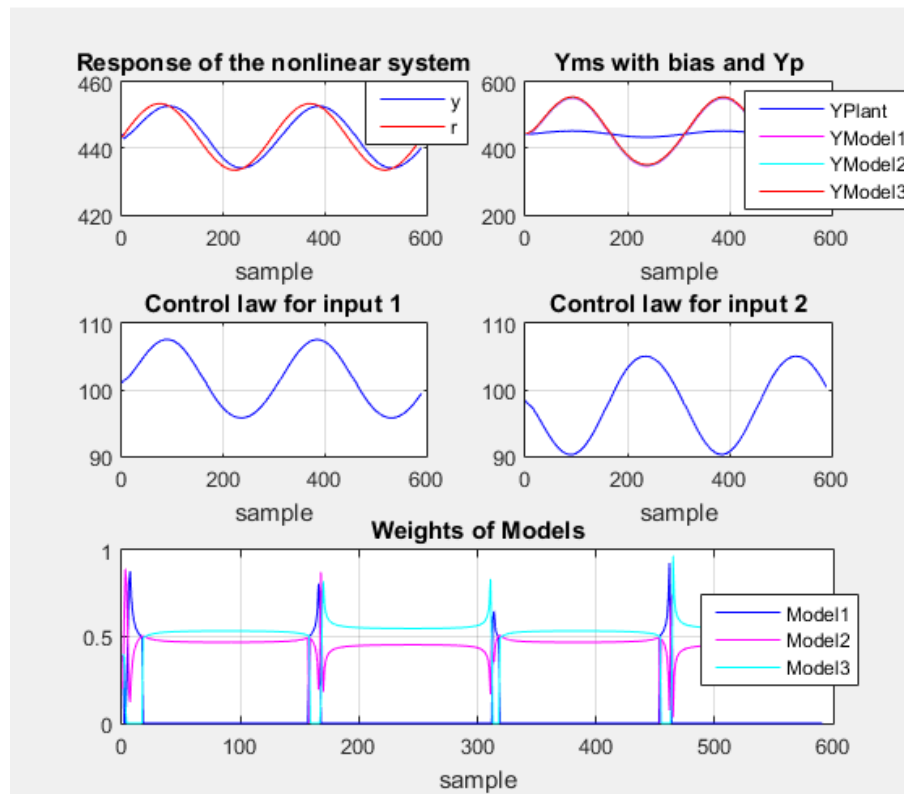


Figure 5. Response of the system with sine wave around operation point of model 2 as a desired output.

پاسخ سیستم به موج سینوسی حول نقطه کار مدل سوم در شکل زیر آورده شده است.

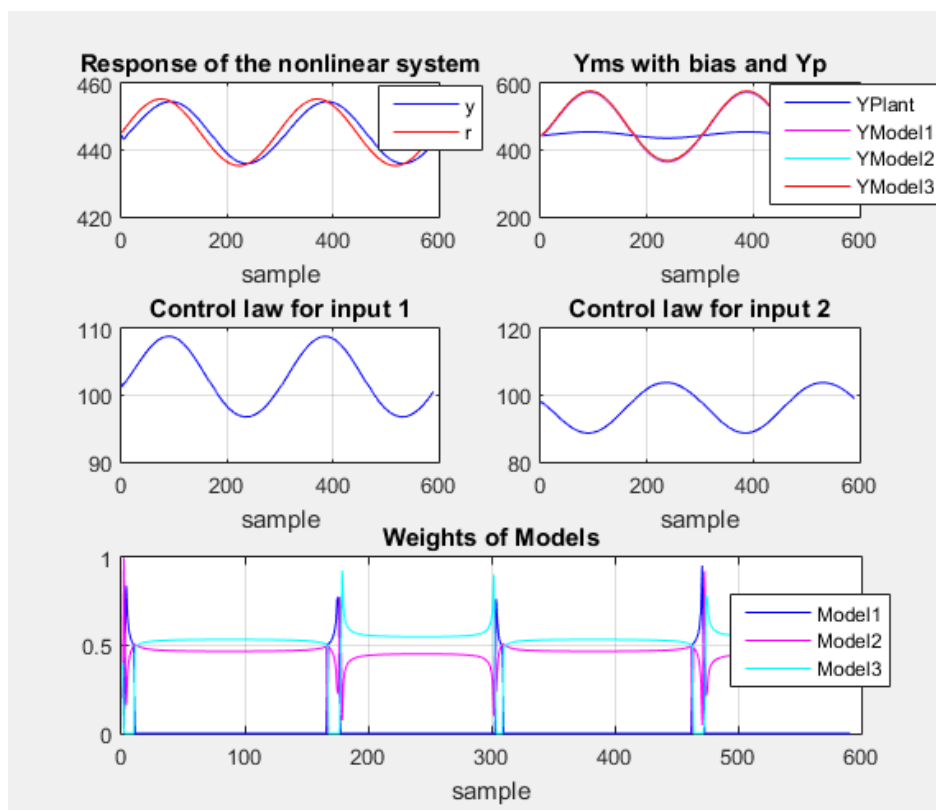


Figure 6. Response of the system with sine wave around operation point of model 3 as a desired output.

همانطور که در هر سه 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 یک موج مربعی با پرش‌ها مختلف را به عنوان خروجی مطلوب به سیستم می‌دهیم. این موج بدین صورت است که دارای 6 پریود است که سه تا از پریودها روی نقاط کار و 3 تا از آنها روی مقدار وسط نقاط کارها می‌باشد. پاسخ سیستم به ازای این خروجی مطلوب در شکل زیر آورده شده است.

همانطور که در این شکل می‌بینیم پاسخ سیستم خروجی مطلوب را دنبال می‌کند.

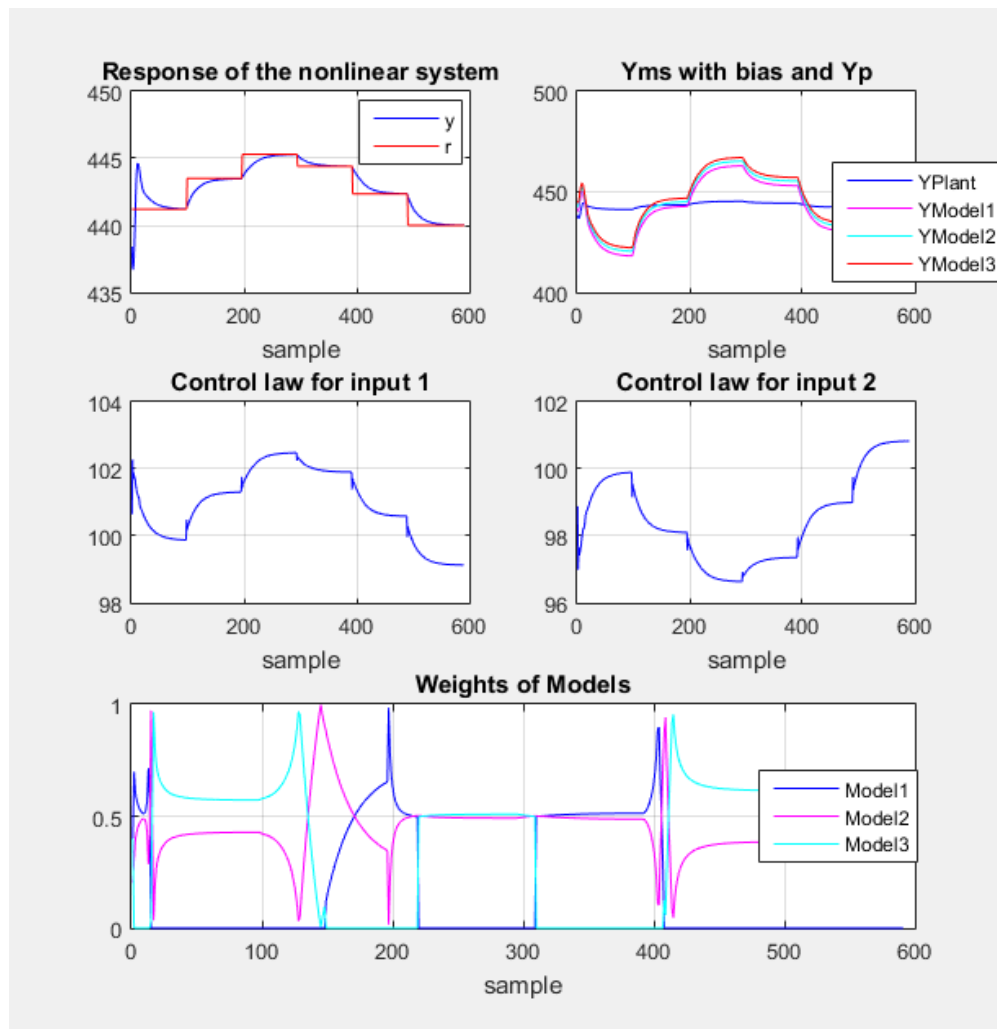


Figure 7. Response of the system with step with various jumps as a desired output.

همانطور که مشاهده می‌کنیم پاسخ سیستم خروجی مطلوب را دنبال می‌کند.

3.3. خروجی مطلوب موج مربعی:

در این قسمت پاسخ سیستم به ازای خروجی مطلوب موج مربعی را رسم می‌کنیم. این خروجی مطلوب را به ازای دامنه 10 حول سه نقطه کار رسم می‌کنیم.

ابتدا پاسخ سیستم به ازای موج مربعی حول نقطه کار مدل اول رسم می‌کنیم و این پاسخ در شکل زیر آورده شده است. همانطور که در این شکل می‌بینیم پاسخ سیستم خروجی مطلوب را دنبال می‌کند.

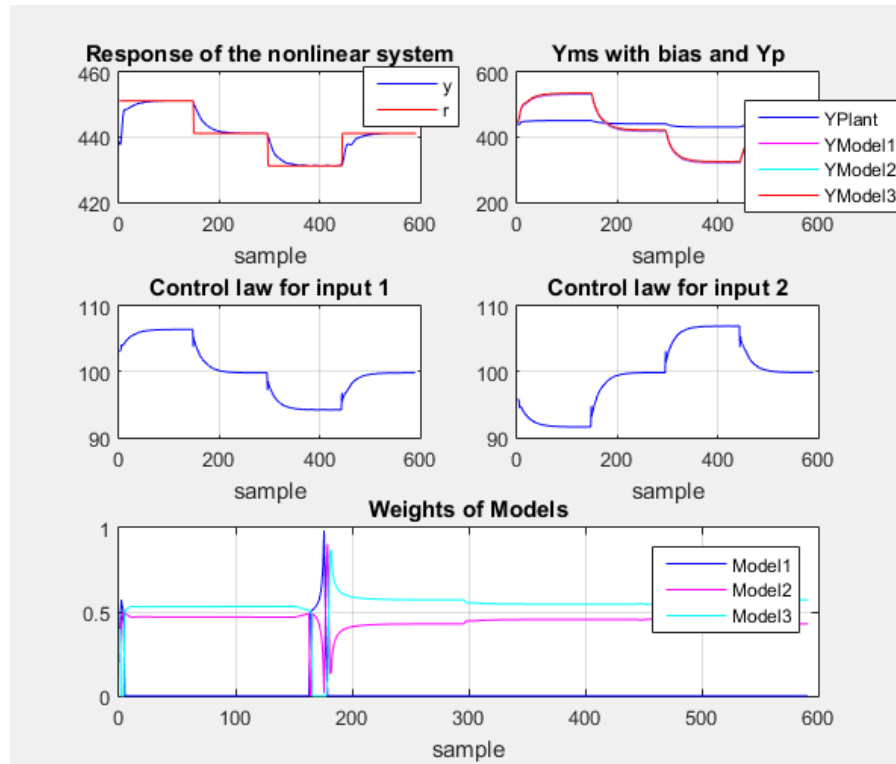


Figure 8. Response of the system with square wave around operation point of model 1 as a desired output.

پاسخ سیستم به موج مربعی حول نقطه کار مدل دوم در شکل زیر آورده شده است.

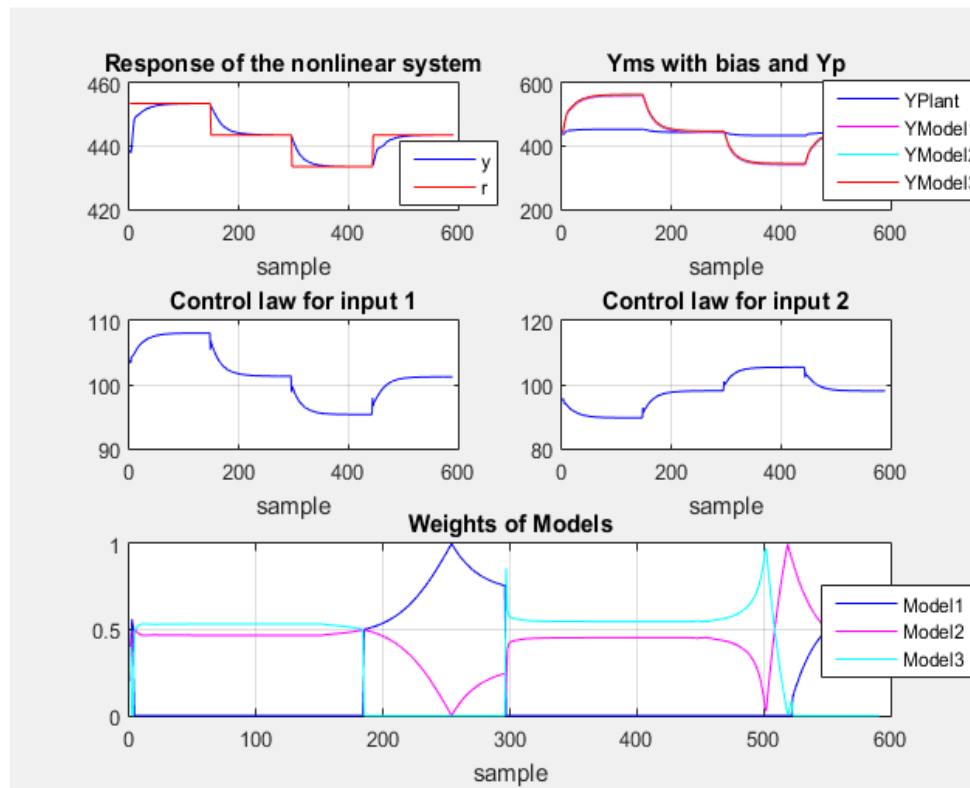


Figure 9. Response of the system with square wave around operation point of model 2 as a desired output.

پاسخ سیستم به موج مربعی حول نقطه کار مدل سوم در شکل زیر آورده شده است.

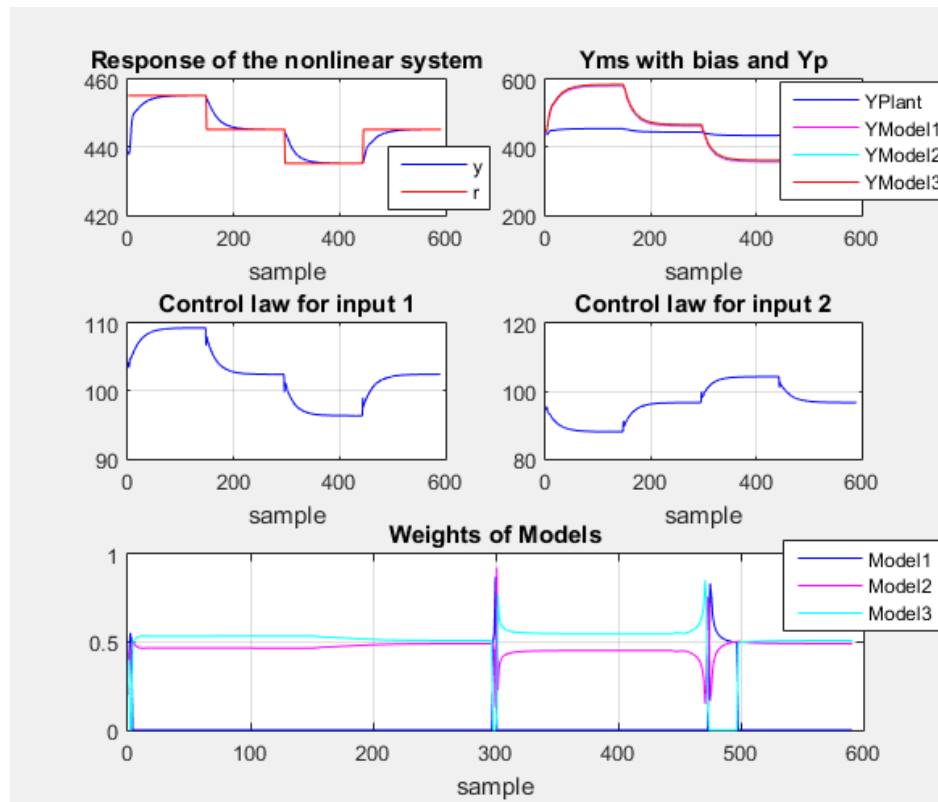


Figure 10. Response of the system with square wave around operation point of model 3 as a desired output.

4. بررسی اثر پارامترها:

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

4.1. بررسی پارامتر گاما (ماتریس R):

در این قسمت به بررسی پارامتر گاما می پردازیم با توجه به اینکه سه مدل داریم این بررسی را به ازای تغییر گامای هر کدام از گاماها جداگانه بررسی می کنیم.

گاما را در هر قسمت برابر با 1، 1/2 و 1/60 قرار می دهیم.

در بررسی گامای هر مدل گامای سایر مدل ها یک قرار داده شده است.

شکل زیر مربوط به بررسی گامای مدل یک است. همانطور که در این شکل می بینیم چون دامنه خروجی مطلوب به نقطه کار مدل دو و سه نزدیک تر است این دو مدل در اکثر زمان ها روشن شده اند. در نتیجه چون مدل یک در اکثر زمان ها خاموش است تغییر گامای مربوط به آن تاثیری روی پاسخ نگذاشته است.

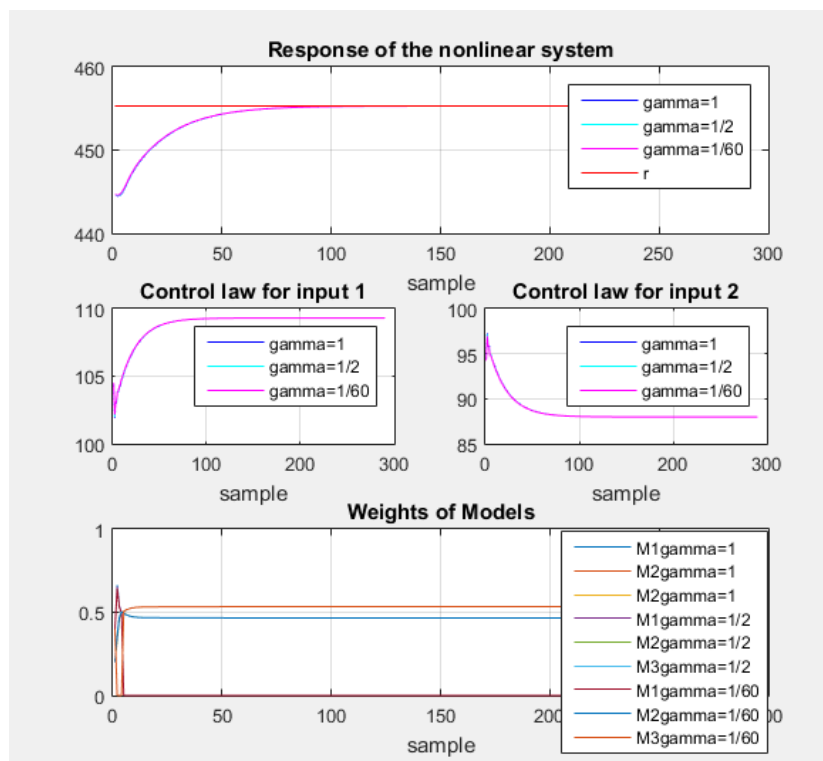


Figure 11. Responses of the system in verifying R for Model 1.

شکل زیر پاسخ سیستم به ازای تغییرات گامای مدل دوم می‌باشد.

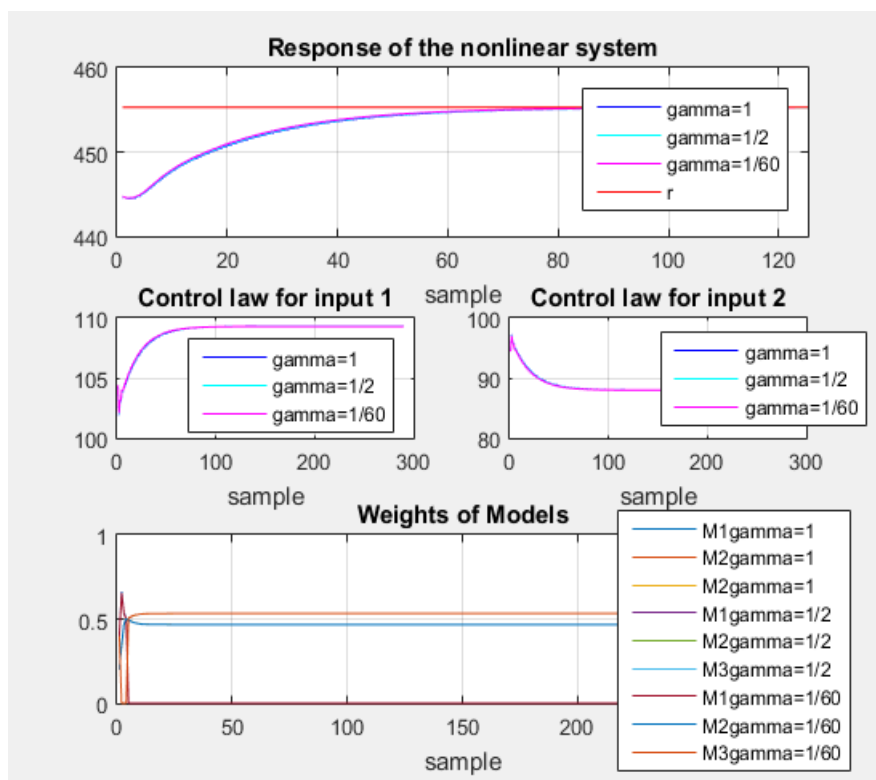


Figure 12. Responses of the system in verifying R for Model 2.

در شکل بالا می‌توان دید که کاهش گامای مدل دو 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. گرچه این تاثیر بسیار کم است ولی وقتی روی عکس زوم کنیم می‌توانیم این تندتر شدن را بهتر ببینیم.

شکل زیر پاسخ سیستم به ازای تغییرات گامای مدل سوم می‌باشد.

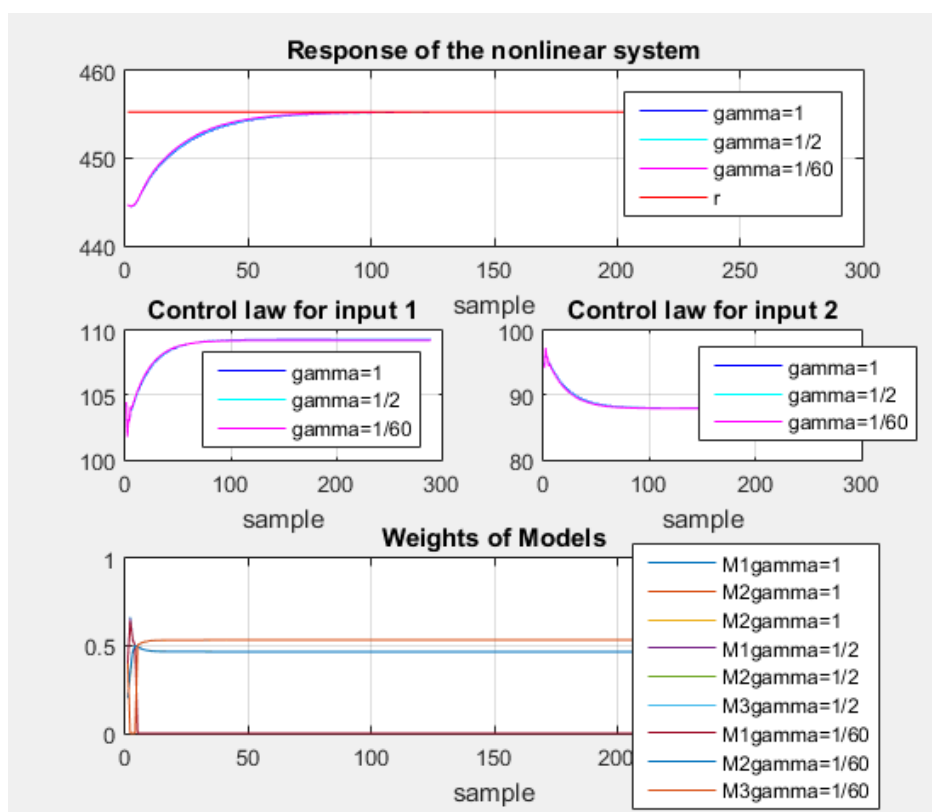


Figure 13. Responses of the system in verifying R for Model 3.

این حالت نیز مشابه مدل 2 می‌باشد که کاهش گاما تا حدودی باعث تندتر شدن پاسخ سیستم شده است.

با توجه به اینکه تاثیر تغییر گامای هر مدل به تنهایی نامحسوس بود و به وضوح دیده نمی‌شد ما در این مرحله تاثیر تغییر گامای همه‌ی مدل‌ها را با هم بررسی می‌کنیم. بدین معنی که گامای سه مدل را با هم برابر قرار داده و آن را به ازای همان سه مقدار گفته شده تغییر می‌دهیم.

نتایج تغییر گامای سه مدل با هم در شکل زیر آورده شده است. همانطور که می‌بینیم خطای حالت ماندگار با تغییر گاما تغییری نمی‌کند و همچنان صفر است اما پاسخ حالت گذرا با کاهش گاما تا حدودی تندتر شده است.

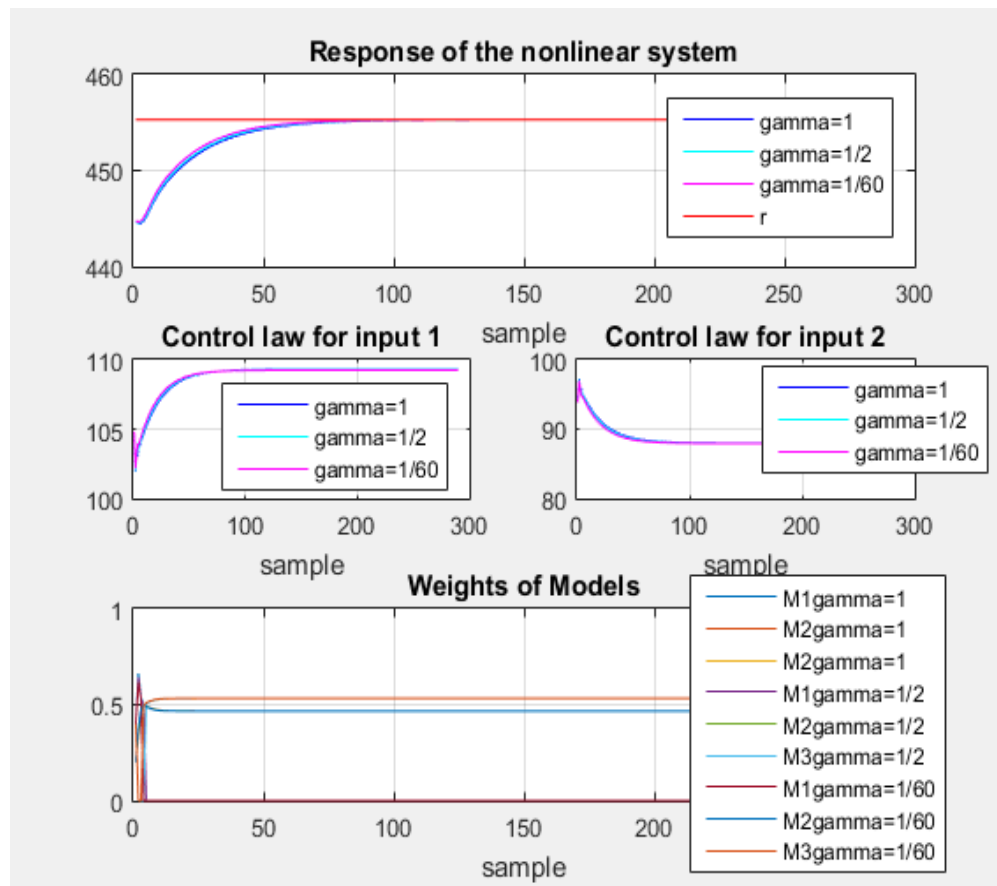


Figure 14. Responses of the system in verifying R .

4.2. بررسی تغییر ماتریس Q :

در بخش قبل دیدیم که وقتی گامای هر مدل را تغییر می‌دادیم تاثیر روی پاسخ به وضوح دیده نمی‌شد در نتیجه این قسمت فقط تغییرات Q همه‌ی مدل‌ها با هم را بررسی می‌کنیم. یعنی Q همه مدل‌ها را برابر با هم قرار می‌دهیم و آن را تغییر می‌دهیم. بنابراین در این قسمت تاثیر تغییر ماتریس Q به ازای مقادیر $Q=I$, $Q=2I$ و $Q=60I$ را بررسی می‌کنیم.

شکل زیر پاسخ سیستم به ازای این تغییرات را نشان می‌دهد.

همانطور که در این شکل می‌بینیم تاثیر Q برعکس تاثیر R است یعنی می‌بینیم که افزایش ماتریس Q پاسخ سیستم را تا حدودی تندتر کرده است که این موضوع در R برعکس بود. لازم به ذکر است که افزایش Q تاثیری بر روی پاسخ حالت ماندگار سیستم نگذاشته است و همچنان خطای حالت ماندگار صفر است.

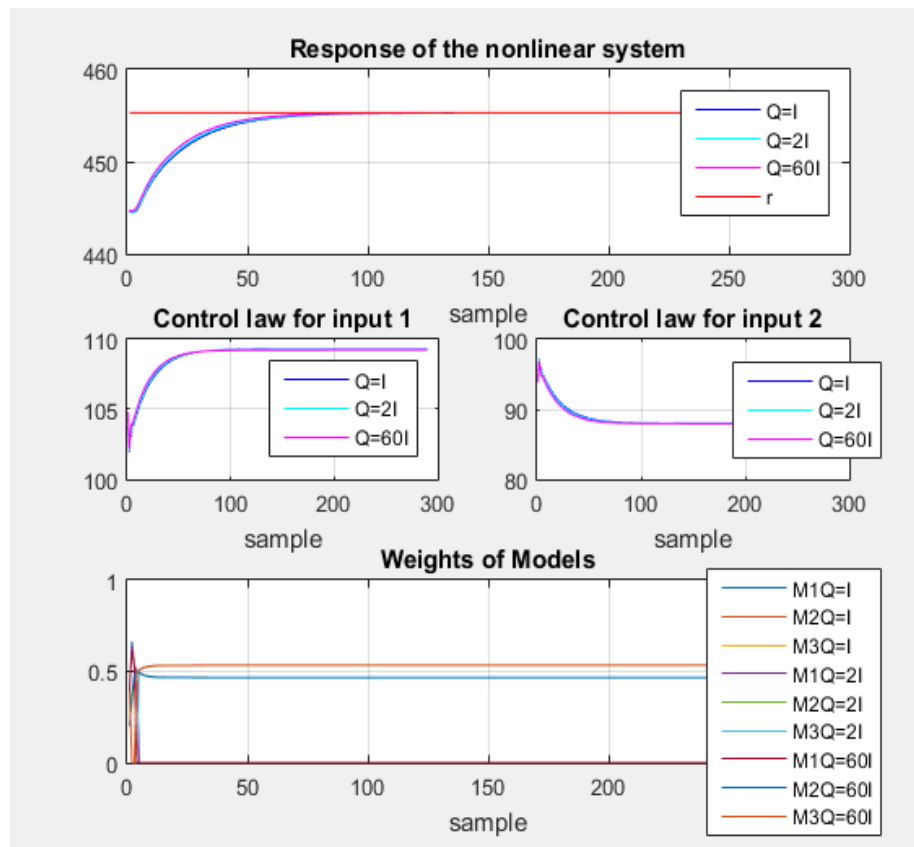


Figure 15. Responses of the system in verifying Q .

4.3. بررسی پارامتر α :

با توجه به اینکه قسمت مرجع کل سیستم یکی است در نتیجه α برای کل سیستم یکی بیشتر نیست و در واقع یک فیلتر بیشتر وجود ندارد که آن هم سیگنال مرجع کل را فیلتر می‌کند. بنابراین ما در این بخش به بررسی اثر پارامتر α بر روی پاسخ سیستم می‌پردازیم.

این بررسی را به ازای α های 0.1، 0.5 و 0.8 بررسی شده است.

پاسخ سیستم به ازای تغییرات مذکور در α در شکل زیر آورده شده است. همانطور که در این شکل می‌بینیم افزایش α باعث کند شدن پاسخ سیستم شده است. بنابراین می‌توان با α های کوچکتر به پاسخ سریع‌تری رسید.

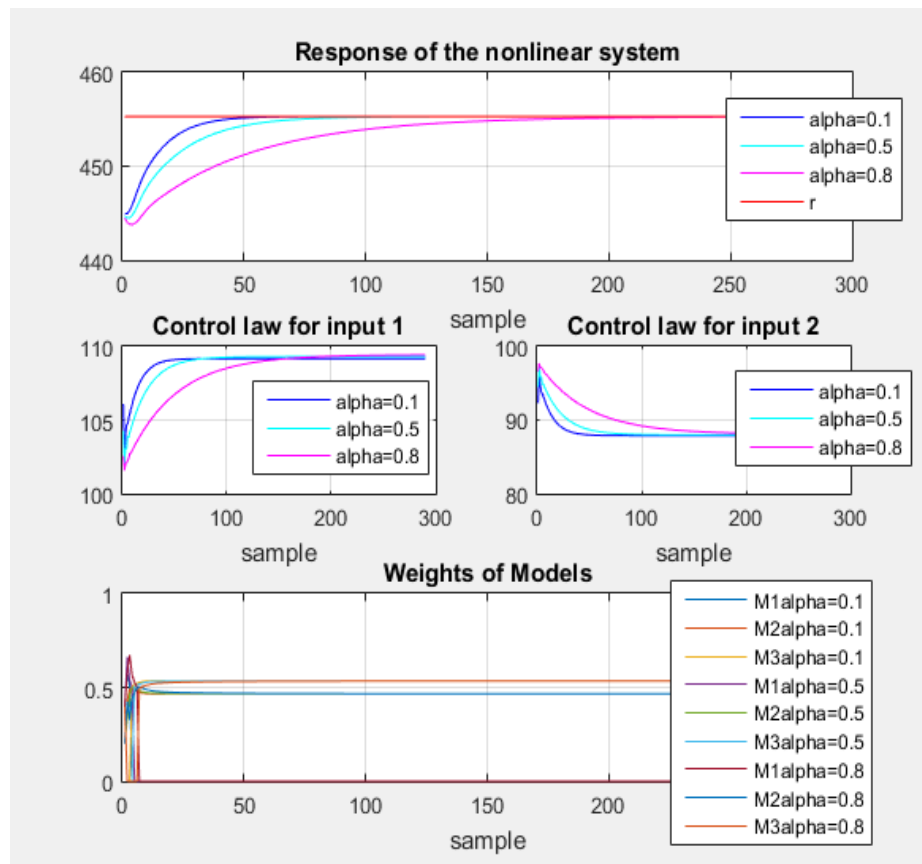


Figure 16. Responses of the system in verifying α .

4.4. بررسی پارامتر P :

در این بخش به بررسی تاثیر پارامتر P بر روی پاسخ سیستم می‌پردازیم. پاسخ سیستم را به ازای P های 2، 5 و 15 رسم شده است. شکل 17 پاسخ سیستم به ازای P های مختلف را نشان می‌دهد. لازم به ذکر است که مقدار M برابر با 5 قرار داده شده است.

همانطور که در شکل 17 می‌بینیم تغییر پارامتر P تاثیر محسوسی 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.

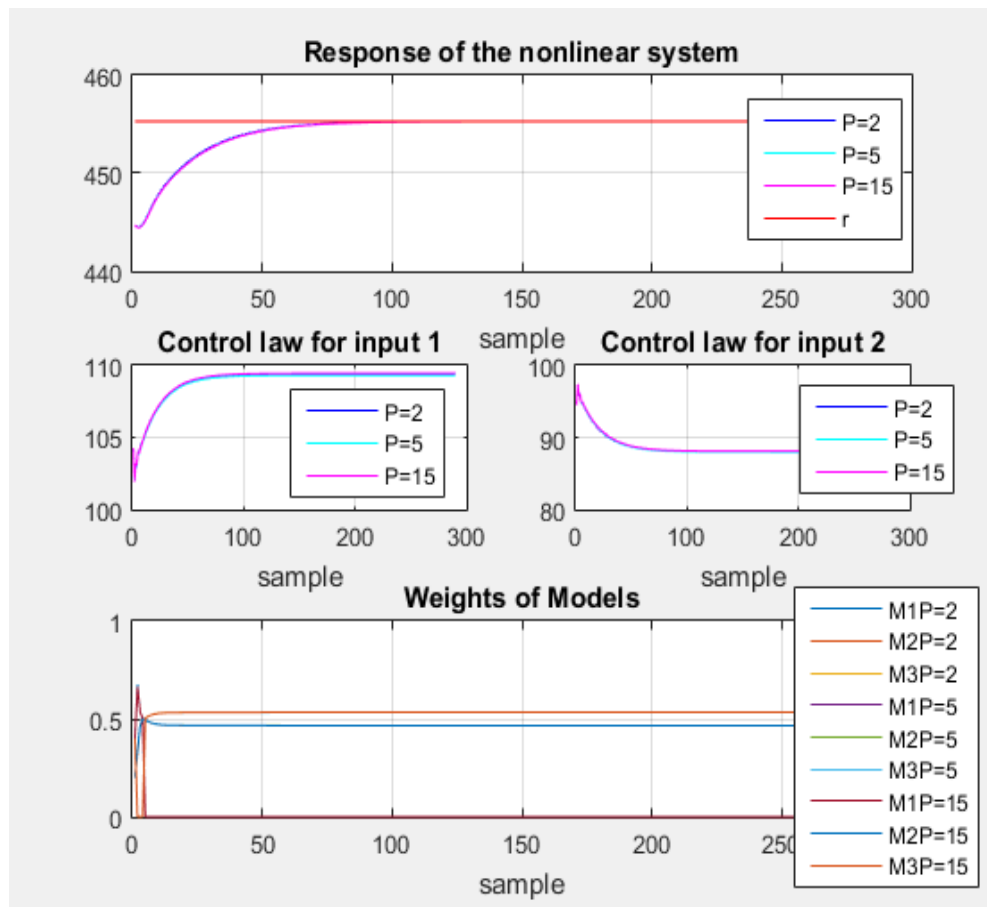


Figure 17. Responses of the system in verifying P .

4.5. بررسی پارامتر M :

در این بخش به بررسی تاثیر پارامتر M بر روی پاسخ سیستم می‌پردازیم. پاسخ سیستم را به ازای M های 2، 5 و 15 رسم شده است. شکل 18 پاسخ سیستم به ازای M های مختلف را نشان می‌دهد. لازم به ذکر است که مقدار P برابر با 5 قرار داده شده است.

همانطور که در شکل 18 می‌بینیم تغییر پارامتر M تاثیر محسوسی 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.

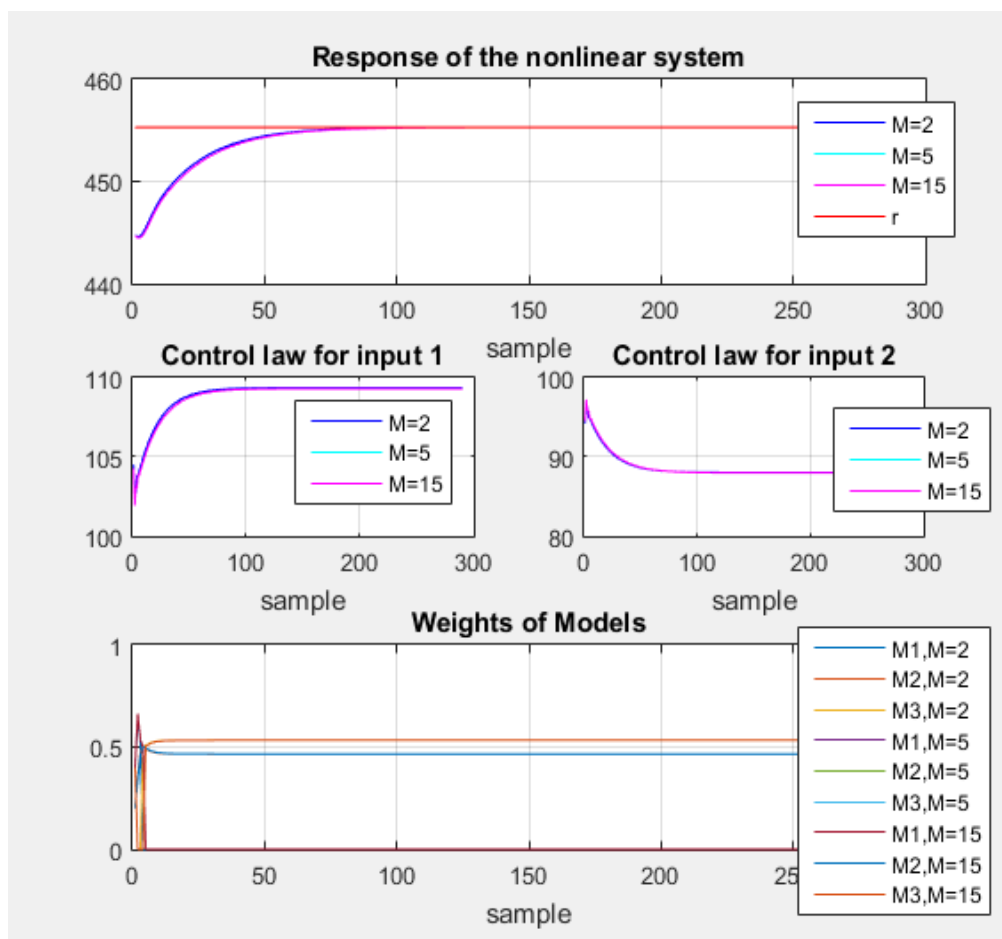


Figure 18. Responses of the system in verifying M .

4.6. بررسی پارامتر T_s :

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

همانطور که در شکل زیر می بینیم وقتی زمان نمونه برداری را کم می کنیم باعث تند شدن پاسخ سیستم می شود. اما این تند شدن پاسخ تا جایی ادامه دارد یعنی از یک مقداری به بعد دیگر کاهش T_s تاثیر چندانی در سریع شدن پاسخ ندارد.

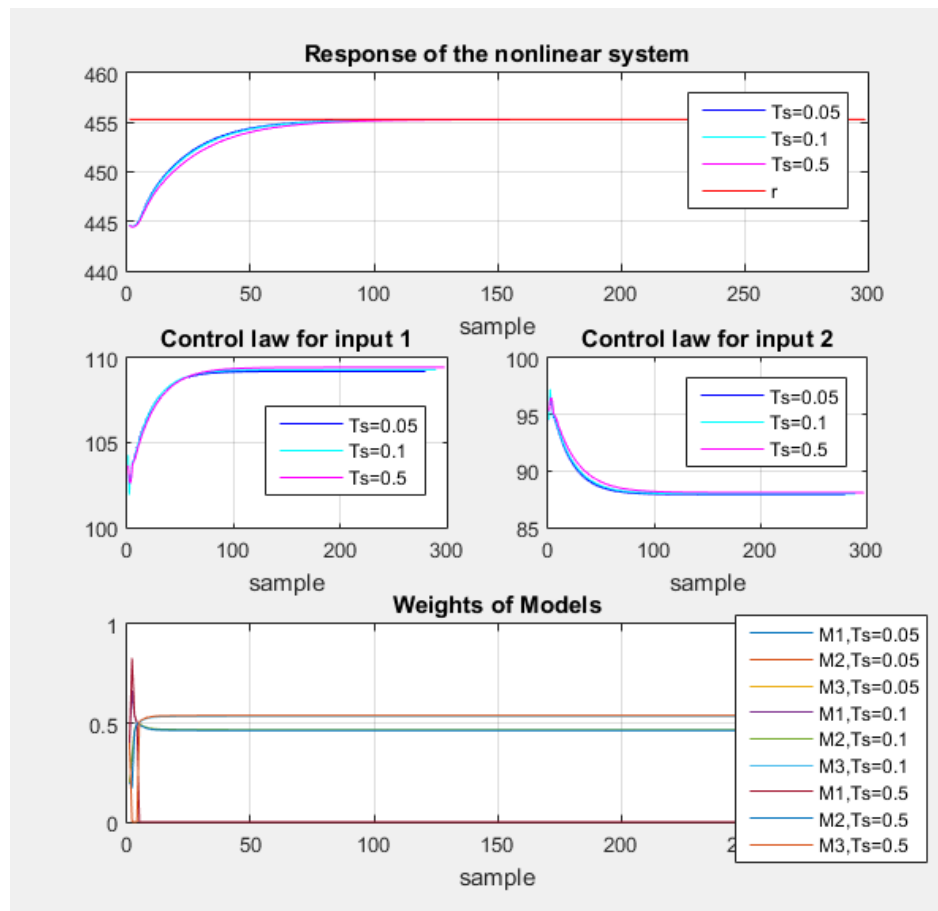


Figure 19. Responses of the system in verifying T_s .

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.

5.1. بررسی اثر اغتشاش:

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

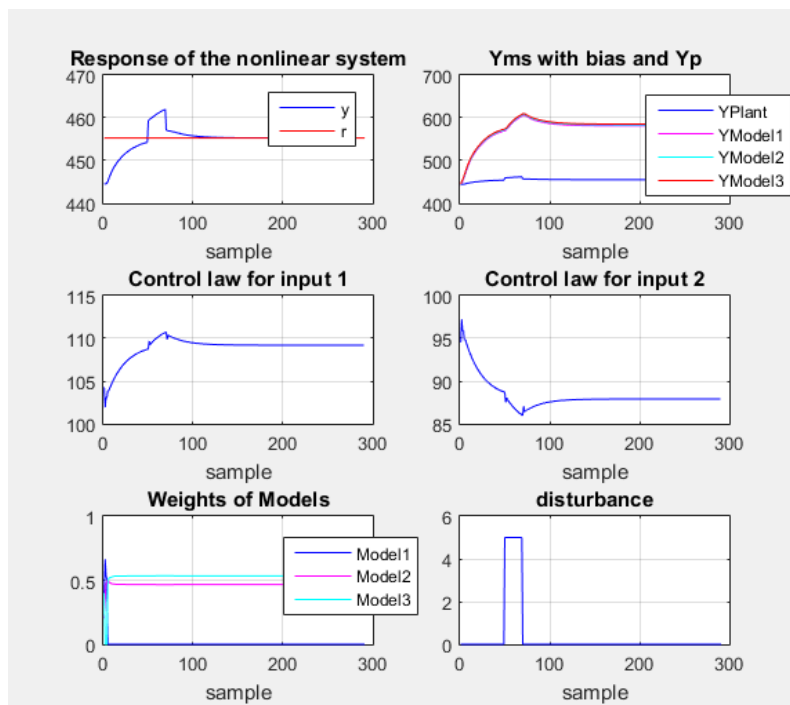


Figure 20. Responses of the system in verifying the effect of disturbance.

5.2. بررسی اثر نویز:

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

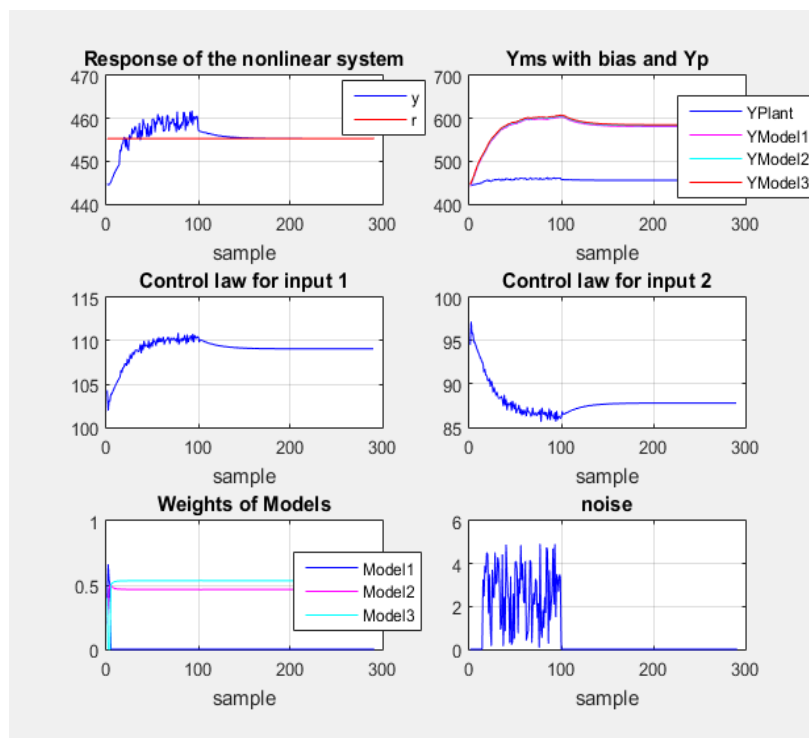


Figure 21. Responses of the system in verifying the effect of noise.

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 5% و 1% اضافه می کنیم. پاسخ سیستم به ازای این عدم قطعیت ها برای این پارامتر در شکل زیر آورده شده است.

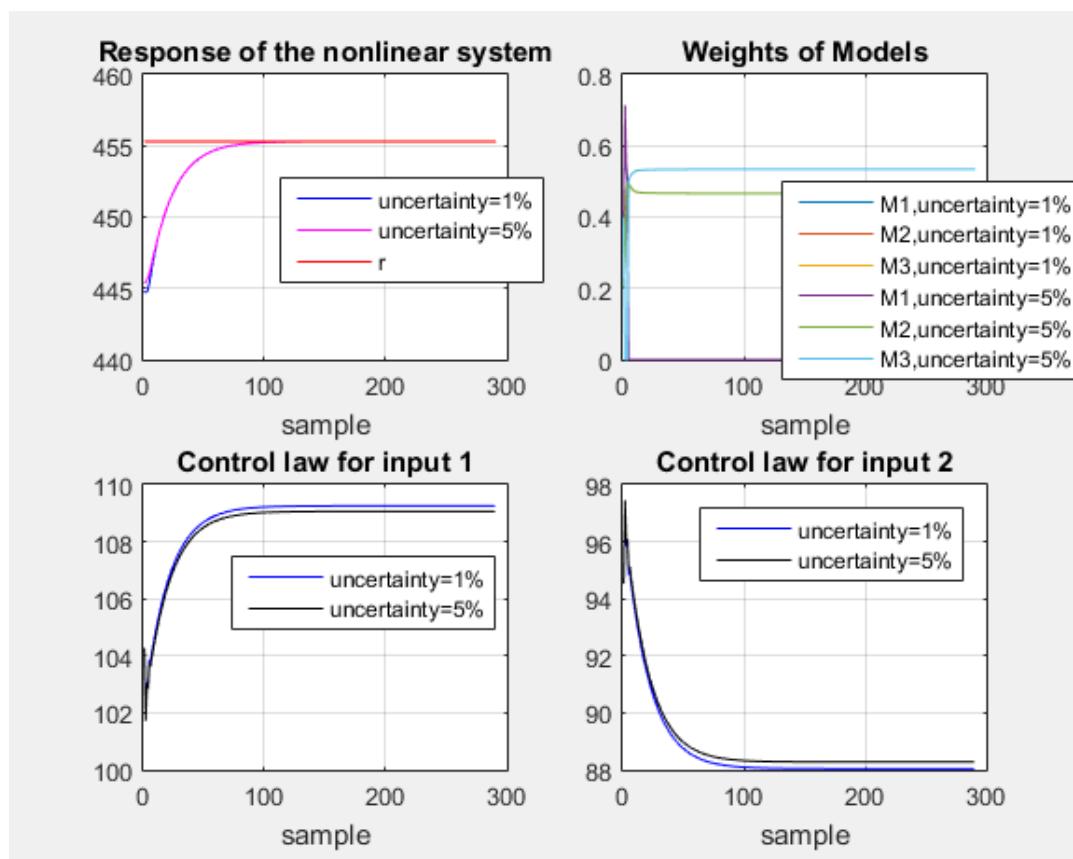


Figure 22. Responses of the system in verifying the effect of uncertainty on V .

همانطور که در شکل بالا می بینیم عدم قطعیت با وجود این کنترل کننده تاثیر چندانی روی پاسخ سیستم نمی گذارد. می توان برتری این کنترل کننده نسبت به کنترل کننده های قبلی را در مقاوم بودن نسبت به عدم قطعیت دید.

اینبار عدم قطعیت با درصد های 10% و 20% را به پارامتر C_{A0} اضافه می کنیم. شکل زیر پاسخ سیستم در حضور عدم قطعیت های مذکور را نشان می دهد.

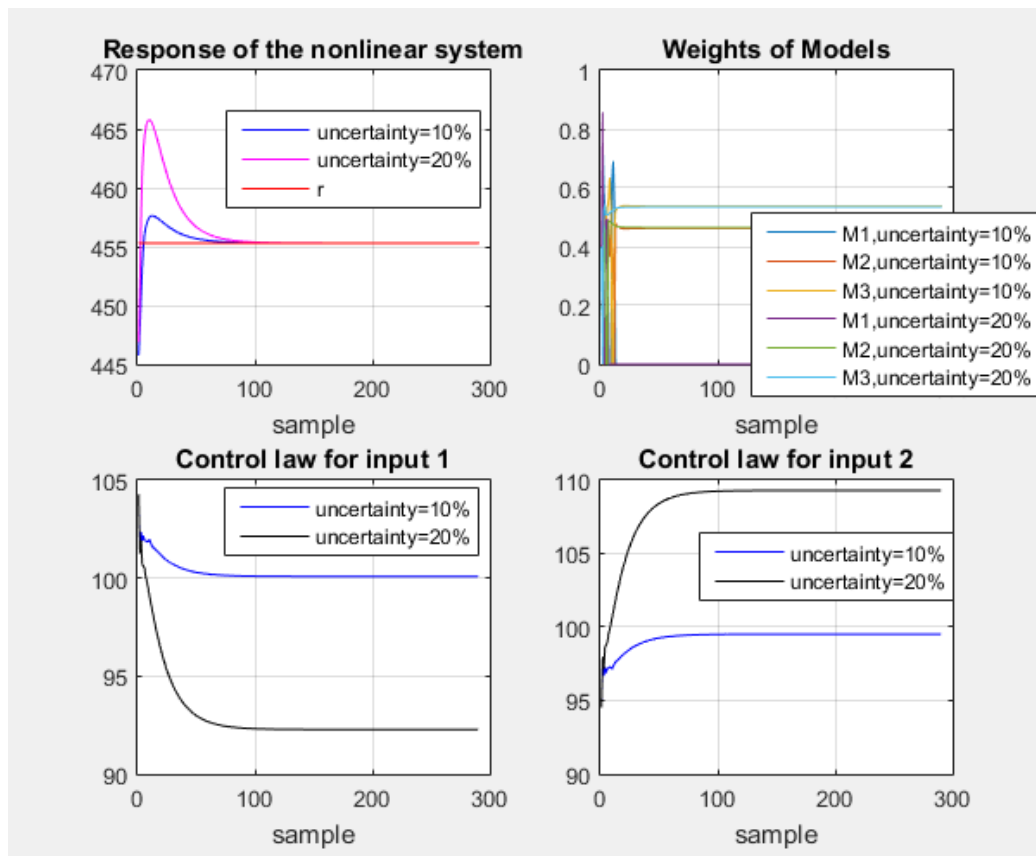


Figure 23. Responses of the system in verifying the effect of uncertainty on CA_0 .

همانطور که در شکل بالا می بینیم حتی با وجود عدم قطعیت های 10% و 20% تنها پاسخ حالت گذرا تا حدودی تحت تاثیر قرار گرفته است و خطای حالت ماندگار را به صفر رسیده است.

7. کدها و شبیه سازی ها:

کدها و شبیه سازی های مورد استفاده در این پروژه در زیر آورده شده است. کد زیر یک فانکشن است که به منظور فراخوانی سه 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
end

```

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

```

clear
clc
%% .....First Model.....
%% .....
[n11,d11,n21,d21]=Inputsys(1);
Gs11 = tf(n11,d11);
Ts=0.1;

```

```

Gd11 = c2d(Gs11,Ts,'zoh');
[num11,den11]=tfdata(Gd11,'v');
Gs21 = tf(n21,d21);
Gd21 = c2d(Gs21,Ts,'zoh');
[num21,den21]=tfdata(Gd21,'v');
sys_info = stepinfo(Gd11);
ts11 = sys_info.SettlingTime;
tr11=sys_info.RiseTime;
sys_info = stepinfo(Gd21);
ts21 = sys_info.SettlingTime;
tr21=sys_info.RiseTime;
t=1:Ts:30;
[g11,t11] = step(Gd11,t);
[g21,t21] = step(Gd21,t);
P11=floor(tr11/Ts);
P21=floor(tr21/Ts);
N11=floor( ts11/Ts);
N21=floor( ts21/Ts);
P1=max(P11,P21);
N=max(N11,N21);
M1=P1;
%.....Second.....
[n12,d12,n22,d22]=Inputs(2);
Gs12 = tf(n12,d12);
Gd12 = c2d(Gs12,Ts,'zoh');
[num12,den12]=tfdata(Gd12,'v');
Gs22 = tf(n22,d22);
Gd22 = c2d(Gs22,Ts,'zoh');
[num22,den22]=tfdata(Gd22,'v');
sys_info = stepinfo(Gd12);
ts12 = sys_info.SettlingTime;
tr12=sys_info.RiseTime;
sys_info = stepinfo(Gd22);
ts22 = sys_info.SettlingTime;
tr22=sys_info.RiseTime;
[g12,t1] = step(Gd12,t);
[g22,t2] = step(Gd22,t);
P12=floor(tr12/Ts);
P22=floor(tr22/Ts);
N12=floor( ts12/Ts);
N22=floor( ts22/Ts);
P2=max(P12,P22);
N2=max(N12,N22);
M2=P2;
%.....Third.....
[n13,d13,n23,d23]=Inputs(3);
Gs13 = tf(n13,d13);
Gd13 = c2d(Gs13,Ts,'zoh');
[num13,den13]=tfdata(Gd13,'v');
Gs23 = tf(n23,d23);
Gd23 = c2d(Gs23,Ts,'zoh');
[num23,den23]=tfdata(Gd23,'v');
sys_info = stepinfo(Gd13);
ts13 = sys_info.SettlingTime;
tr13=sys_info.RiseTime;
sys_info = stepinfo(Gd23);

```



```

ts23 = sys_info.SettlingTime;
tr23=sys_info.RiseTime;
[g13,t13] = step(Gd13,t);
[g23,t23] = step(Gd23,t);
P13=floor(tr13/Ts);
P23=floor(tr23/Ts);
N13=floor( ts13/Ts);
N23=floor( ts23/Ts);
P3=max(P13,P23);
N3=max(N13,N23);
M3=P3;
P=max(M1,M2);
P=max(P,M3);
M=P;
P3=P; P2=P; P1=P;
M3=M; M2=M; M1=M;
% .....
% % .....First Model.....
% % .....
% .....Toeplitz Matrix.....
b11 = zeros(1,P1); b11(1,1)= g11(2);
a11 = g11(2:P1+1);
G11 = toeplitz(a11,b11);
G11(:,M1) = G11(:,M1:P1)*ones(P1-M1+1,1);
G11 = G11(:,1:M1);
% .....
b21 = zeros(1,P1); b21(1,1)= g21(2);
a21 = g21(2:P1+1);
G21 = toeplitz(a21,b21);
G21(:,M1) = G21(:,M1:P1)*ones(P1-M1+1,1);
G21 = G21(:,1:M1);
G1=[G11 G21];
% .....
% A1~=-1-2.564z^-1+2.2365z^-2-0.6725z^-3
% According to the discrete transfer function, below parameters have been
% defined
na=3;
nb1=1; nb2=1;
nb=nb1;
d=0;
N11=d+1;
N21=d+P1;
% .....
a1_=[1 -2.564 2.2365 -0.6725];
% .....
b11_=num11(2:end);
b21_=num21(2:end);
C=1; % because of using white noise
f1=zeros(P1+d,na+1);
f1(1,1:3)=-1*a1_(2:4);
for j=1:P1+d-1
    for i=1:na
        f1(j+1,i)=f1(j,i+1)-f1(j,1)*a1_(i+1);
    end
end
F1=f1(N11:N21,1:na);

```

```

% .....
E11=zeros(P1);
E11(:,1)=1;
for j=1:P1-1
    E11(j+1:P1,j+1)=f1(j,1);
end
B11=zeros(P1,P1+nb);
for k=1:P1
    B11(k,k:k+1)=b11_;
end
m11_=E11*B11;
M11_=zeros(P1,nb+d);
for k=1:P1
    M11_(k,:)=m11_(k,k+1);
end
% .....
E21=zeros(P1);
E21(:,1)=1;
for j=1:P1-1
    E21(j+1:P1,j+1)=f1(j,1);
end
B21=zeros(P1,P1+nb);
for k=1:P1
    B21(k,k:k+1)=b21_;
end
m21_=E21*B21;
M21_=zeros(P1,nb+d);
for k=1:P1
    M21_(k,:)=m21_(k,k+1);
end
M1_=[M11_ M21_];
% .....
gamma =1;
gain_DC11=(num11(1)+num11(2)+num11(3))/(den11(1)+den11(2)+den11(3));
gain_DC21=(num21(1)+num21(2)+num21(3))/(den21(1)+den21(2)+den21(3));
Q1 = eye(P1);
R11 =((1.2)^2)*gamma*gain_DC11^2*eye(M1);
R21=gamma*gain_DC21^2*eye(M1);
R1=[R11 zeros(M1); zeros(M1) R21];
alpha=0.5;
Kgpc1=(G1'*Q1*G1+R1)\(G1'*Q1);
%% .....
%% .....Second
Model.....
%% .....
% .....Toeplitz Matrix.....
b12 = zeros(1,P2); b12(1,1)= g12(2);
a12 = g12(2:P2+1);
G12 = toeplitz(a12,b12);
G12(:,M2) = G12(:,M2:P2)*ones(P2-M2+1,1);
G12 = G12(:,1:M2);
% .....
b22 = zeros(1,P2); b22(1,1)= g22(2);
a22 = g22(2:P2+1);

```

```

G22 = toeplitz(a22,b22);
G22(:,M2) = G22(:,M2:P2)*ones(P2-M2+1,1);
G22 = G22(:,1:M2);
G2=[G12 G22];
% .....
% A2~=-1-2.411z^-1+1.9514z^-2-0.5404z^-3
% According to the discrete transfer function, below parameters have been
% defined
N12=d+1;
N22=d+P2;
% .....
a2_=[1 -2.474 2.0667 -0.5927];
% .....
b12_=num12(2:end);
b22_=num22(2:end);
C=1; % because of using white noise
f2=zeros(P2+d,na+1);
f2(1,1:3)=-1*a2_(2:4);
for j=1:P2+d-1
    for i=1:na
        f2(j+1,i)=f2(j,i+1)-f2(j,1)*a2_(i+1);
    end
end
F2=f2(N12:N22,1:na);
% .....
E12=zeros(P2);
E12(:,1)=1;
for j=1:P2-1
    E12(j+1:P2,j+1)=f2(j,1);
end
B12=zeros(P2,P2+nb);
for k=1:P2
    B12(k,k:k+1)=b12_;
end
m12_=E12*B12;
M12_=zeros(P2,nb+d);
for k=1:P2
    M12_(k,:)=m12_(k,k+1);
end
% .....
E22=zeros(P2);
E22(:,1)=1;
for j=1:P2-1
    E22(j+1:P2,j+1)=f2(j,1);
end
B22=zeros(P2,P2+nb);
for k=1:P2
    B22(k,k:k+1)=b22_;
end
m22_=E22*B22;
M22_=zeros(P2,nb+d);
for k=1:P2
    M22_(k,:)=m22_(k,k+1);
end
M2_=[M12_ M22_];
% .....

```

```

gamma2 =1;
gain_DC12=(num12(1)+num12(2)+num12(3))/(den12(1)+den12(2)+den12(3));
gain_DC22=(num22(1)+num22(2)+num22(3))/(den22(1)+den22(2)+den22(3));
Q2 = eye(P2);
R12 =((gain_DC22/gain_DC12)^2)*gamma2*gain_DC12^2*eye(M2);
R22=gamma2*gain_DC22^2*eye(M2);
R2=[R12 zeros(M2); zeros(M2) R22];
%alpha2=0.5;
Kgpc2=(G2'*Q2*G2+R2)\(G2'*Q2);
%% .....
%% .....Third
Model.....
%% .....
%.....Toeplitz Matrix.....
b13 = zeros(1,P3); b13(1,1)= g13(2);
a13 = g13(2:P3+1);
G13 = toeplitz(a13,b13);
G13(:,M3) = G13(:,M3:P3)*ones(P3-M3+1,1);
G13 = G13(:,1:M3);
% .....
b23 = zeros(1,P3); b23(1,1)= g23(2);
a23 = g23(2:P3+1);
G23 = toeplitz(a23,b23);
G23(:,M3) = G23(:,M3:P3)*ones(P3-M3+1,1);
G23 = G23(:,1:M3);
G3=[G13 G23];
% .....
% A3~=-1-2.725z^-1+2.5357z^-2-0.8107z^-3
% According to the discrete transfer function, below parameters have been
% defined
N13=d+1;
N23=d+P3;
% .....
a3_=[1 -2.411 1.9514 -0.5404];%[1 -2.725 2.5357 -0.8107];
% .....
b13_=num13(2:end);
b23_=num23(2:end);
C=1; % because of using white noise
f3=zeros(P3+d,na+1);
f3(1,1:3)=-1*a3_(2:4);
for j=1:P3+d-1
    for i=1:na
        f3(j+1,i)=f3(j,i+1)-f3(j,1)*a3_(i+1);
    end
end
F3=f3(N13:N23,1:na);
% .....
E13=zeros(P3);
E13(:,1)=1;
for j=1:P3-1
    E13(j+1:P3,j+1)=f3(j,1);
end
B13=zeros(P3,P3+nb);
for k=1:P3

```

[illegible]

[illegible]

```

U22=zeros(M2,length(t));
Y2_=zeros(na,length(t));
%.....Third.....
dU13_=zeros(nb+d,length(t));
dU23_=zeros(nb+d,length(t));
dU3_=[dU13_;dU23_];
d3=zeros(1,length(t));
%.....
ym3=[];
y3=0;
Y_d3=zeros(P3,length(t));
Y_past3=zeros(P3,length(t));
Y_m3=zeros(P3,length(t));
D3=zeros(P3,length(t));
E3=zeros(P3,length(t));
dU13=zeros(M3,length(t));
dU23=zeros(M3,length(t));
dU3=[dU13;dU23];
U13=zeros(M3,length(t));
U23=zeros(M3,length(t));
Y3_=zeros(na,length(t));
dU01=zeros(M,length(t));
dU02=zeros(M,length(t));
dU=[dU01; dU02];
%.....
%.....
p=zeros(3,length(t));
p(1:3,2)=[0.4 0.2 0.4]';
p_=zeros(3,length(t));
p__=zeros(3,length(t));
e=zeros(2,length(t),3);
p1=p(1,2);
p2=p(2,2);
p3=p(3,2);
% dist=zeros(length(t),1);
% dist(50:69,1)=5*ones(20,1);
% noise=zeros(length(t),1);
% noise(15:99,1)=5*rand(85,1);

for i=1:length(t)-1

for j=1:P1
    Y_d1(j,i+1)=(alpha^j)*y1+(1-(alpha)^j)*(r(i+1)-441.2); % Programmed
end

Y_past1(:,i+1)=M1_*dU1_(:,i+1)+F1*Y1_(:,i+1);
D1(:,i+1)=d1(i+1)*ones(P1,1);

E1(:,i+1)=Y_d1(:,i+1)-Y_past1(:,i+1)-D1(:,i+1);

dU1(:,i+1)=Kgpc1*E1(:,i+1);
dU11(:,i+1)=dU1(1:M1,i+1);
dU21(:,i+1)=dU1(M1+1:2*M1,i+1);
% U11(1,i+1)=dU11(1,i+1)+U11(1,i);
% U21(1,i+1)=dU21(1,i+1)+U21(1,i);

```

```
dU1(:,i+1)=[dU11(:,i+1);dU21(:,i+1)];
```

```
% .....
```

```
u11=U11(1,i)+100;
```

```
u21=U21(1,i)+100;
```

```
% % .....
```

```
.....
```

```
% % .....Second
```

```
Model.....
```

```
% % .....
```

```
.....
```

```
% .....
```

```
for j=1:P2
```

```
Y_d2(j,i+1)=(alpha^j)*y2+(1-(alpha)^j)*(r(i+1)-443.5); % Programmed
```

```
end
```

```
Y_past2(:,i+1)=M2_*dU2(:,i+1)+F2*Y2(:,i+1);
```

```
D2(:,i+1)=d2(i+1)*ones(P2,1);
```

```
E2(:,i+1)=Y_d2(:,i+1)-Y_past2(:,i+1)-D2(:,i+1);
```

```
dU2(:,i+1)=Kgpc2*E2(:,i+1);
```

```
dU12(:,i+1)=dU2(1:M2,i+1);
```

```
dU22(:,i+1)=dU2(M2+1:2*M2,i+1);
```

```
% U12(1,i+1)=dU12(1,i+1)+U12(1,i);
```

```
% U22(1,i+1)=dU22(1,i+1)+U22(1,i);
```

```
dU2(:,i+1)=[dU12(:,i+1);dU22(:,i+1)];
```

```
% .....
```

```
u12=U12(1,i)+100;
```

```
u22=U22(1,i)+97;
```

```
% .....
```

```
% % .....
```

```
.....
```

```
% % .....Third
```

```
Model.....
```

```
% % .....
```

```
.....
```

```
% .....
```

```
for j=1:P3
```

```
Y_d3(j,i+1)=(alpha^j)*y3+(1-(alpha)^j)*(r(i+1)-445.3); % Programmed
```

```
end
```

```
Y_past3(:,i+1)=M3_*dU3(:,i+1)+F3*Y3(:,i+1);
```

```
D3(:,i+1)=d3(i+1)*ones(P3,1);
```



```

E3(:,i+1)=Y_d3(:,i+1)-Y_past3(:,i+1)-D3(:,i+1);

dU3(:,i+1)=Kgpc3*E3(:,i+1);
dU13(:,i+1)=dU3(1:M3,i+1);
dU23(:,i+1)=dU3(M3+1:2*M3,i+1);
% U13(1,i+1)=dU13(1,i+1)+U13(1,i);
% U23(1,i+1)=dU23(1,i+1)+U23(1,i);
dU3(:,i+1)=[dU13(:,i+1);dU23(:,i+1)];
% .....
% we should give models U which has obtained with 3 models with weights
dU=p(1,i+1)*dU1+p(2,i+1)*dU2+p(3,i+1)*dU3;
dU01(:,i+1)=dU(1:M,i+1);
dU02(:,i+1)=dU(M+1:2*M,i+1);
% .....

Y_m1(:,i+1)=G1*dU(:,i+1)+Y_past1(:,i+1);

dU11_(2:nb+d,i+2)=dU11_(1:nb+d-1,i+1);
dU11_(1,i+2)=dU01(1,i+1);
dU21_(2:nb+d,i+2)=dU21_(1:nb+d-1,i+1);
dU21_(1,i+2)=dU02(1,i+1);
dU1_(:,i+2)=[dU11_(:,i+2);dU21_(:,i+2)];
Y1_(2:na,i+2)=Y1_(1:na-1,i+1);
Y1_(1,i+2)=Y_m1(1,i+1);
% .....
Y_m2(:,i+1)=G2*dU(:,i+1)+Y_past2(:,i+1);

dU12_(2:nb+d,i+2)=dU12_(1:nb+d-1,i+1);
dU12_(1,i+2)=dU01(1,i+1);
dU22_(2:nb+d,i+2)=dU22_(1:nb+d-1,i+1);
dU22_(1,i+2)=dU02(1,i+1);
dU2_(:,i+2)=[dU12_(:,i+2);dU22_(:,i+2)];
Y2_(2:na,i+2)=Y2_(1:na-1,i+1);
Y2_(1,i+2)=Y_m2(1,i+1);
% .....

Y_m3(:,i+1)=G3*dU(:,i+1)+Y_past3(:,i+1);

dU13_(2:nb+d,i+2)=dU13_(1:nb+d-1,i+1);
dU13_(1,i+2)=dU01(1,i+1);
dU23_(2:nb+d,i+2)=dU23_(1:nb+d-1,i+1);
dU23_(1,i+2)=dU02(1,i+1);
dU3_(:,i+2)=[dU13_(:,i+2);dU23_(:,i+2)];
Y3_(2:na,i+2)=Y3_(1:na-1,i+1);
Y3_(1,i+2)=Y_m3(1,i+1);

% .....
u13=U13(1,i)+103;
u23=U23(1,i)+97;
% % .....
u1=p(1,i+1)*u11+p(2,i+1)*u12+p(3,i+1)*u13+dU01(1,i+1);
u2=p(1,i+1)*u21+p(2,i+1)*u22+p(3,i+1)*u23+dU02(1,i+1);
U13(1,i+1)=u1-103;
U23(1,i+1)=u2-97;

```

```

U12(1,i+1)=u1-100;
U22(1,i+1)=u2-97;
U11(1,i+1)=u1-100;
U21(1,i+1)=u2-100;
% .....
sim('Model')
% .....
d1(i+2)=y(end)-Y_m1(1,i+1)-441.2;
d2(i+2)=y(end)-Y_m2(1,i+1)-443.5;
d3(i+2)=y(end)-Y_m3(1,i+1)-445.3;
% .....
e(2:end,i+2,1)=e(1:end-1,i+1,1);
e(1,i+2,1)=abs(d1(i+2));
e(2:end,i+2,2)=e(1:end-1,i+1,2);
e(1,i+2,2)=abs(d2(i+2));
e(2:end,i+2,3)=e(1:end-1,i+1,3);
e(1,i+2,3)=abs(d3(i+2));
% .....
sum=0;
for k=1:3
    sum=1/e(1,i+2,k)+sum;
end
w=zeros(3,1);
for j=1:3
    w(j)=(1/e(1,i+2,j))/sum;
end
% for j=1:3
%     p(j,i+2)=w(j);
% end
[max1,ind1]=max(w);
for j=1:3
    if j==ind1
        w(j)=-200;
    end
end
[max2,ind2]=max(w);
% res=[0.5 0; 0 0.1];
% sum2=0;
% for j=1:3
%     sum2=exp(e(:,i+2,j)*res*e(:,i+2,j))*p(j,i+1)+sum2;
% end
% p_(ind1,i+2)=(exp(e(:,i+2,ind1)*res*e(:,i+2,ind1))*p(j,i+1))/sum2;
% p_(ind2,i+2)=(exp(e(:,i+2,ind2)*res*e(:,i+2,ind2))*p(j,i+1))/sum2;
% delta=0.01;
% if p_(ind1,i+2)>=delta
%     p__(ind1,i+2)=p_(ind1,i+2);
% else
%     p__(ind1,i+2)=delta;
% end
% if p_(ind2,i+2)>=delta
%     p__(ind2,i+2)=p_(ind2,i+2);
% else
%     p__(ind2,i+2)=delta;
% end
% sum3=0;
% for j=1:3

```

```

% sum3=sum3+p__(j,i+2);
% end
% for j=1:3
% p(j,i+2)=p__(j,i+2)/sum3;
% end
p(ind1,i+2)=max1;
p(ind2,i+2)=max2;
for j=1:3
    if j~=ind1 && j~=ind2
        p(j,i+2)=0;
    end
end
sum3=0;
for j=1:3
    sum3=sum3+p(j,i+2);
end
for j=1:3
    p(j,i+2)=p(j,i+2)/sum3;
end
y=y(end);%+noise(i,1);%+dist(i,1);
% .....
y1=y(end)+-441.2; % nonlinear
y2=y(end)-443.5; % nonlinear
y3=y(end)-445.3; % nonlinear
% .....
% .....
% y1=[y1; y+441.2];
% y1=[y1; y+445.3];
ynl=[ynl; y(end)];
% .....
ym1=[ym1; Y_m1(1,i)];
ym2=[ym1; Y_m2(1,i)];
ym3=[ym1; Y_m3(1,i)];
u_1=[u_1; u1];
u_2=[u_2; u2];
x01=x1(end);
x02=x2(end);
p1=[p1; p(1,i+2)];
p2=[p2; p(2,i+2)];
p3=[p3; p(3,i+2)];
x_1=[x_1; x1(end)];

end

figure(1);
subplot(3,2,1);
plot(ynl,'b');
hold on
% .....
plot(r,'r');
% plot(r+445.3,'r');
% plot(r+436.8,'r');
% .....
grid on
legend('y','r');

```

```

title('Response of the nonlinear system');
xlabel('sample');
subplot(3,2,2);
%.....
% plot(y1-441.2,'b');
% plot(y1-445.3,'b');
plot(ynl,'b');
%.....
hold on
plot(ym1+441.2,'m');
grid on
hold on
plot(ym2+443.5,'c');
hold on
plot(ym3+445.3,'r');
xlabel('sample');
title('Yms with bias and Yp');
legend('YPlant','YModel1','YModel2','YModel3');
subplot(3,2,3);
plot(u_1,'b');
grid on
xlabel('sample');
title('Control law for input 1');
subplot(3,2,4);
plot(u_2,'b');
grid on
xlabel('sample');
title('Control law for input 2');
subplot(3,2,5:6);
plot(p1,'b');
hold on
plot(p2,'m');
hold on
plot(p3,'c');
legend('Model1','Model2','Model3');
grid on
xlabel('sample');
title('Weights of Models');
subplot(3,2,6);
% plot(dist,'b');
% grid on
% xlabel('sample');
% title('disturbance');
% plot(noise,'b');
% grid on
% xlabel('sample');
% title('noise');
% figure
% plot(x_1);

% legend('uncertainty=1%','uncertainty=5%','r');
% figure(1);
% subplot(2,2,1);
% hold on
% plot(ynl,'m');
% hold on

```

```

% plot(r,'r')
% legend('uncertainty=10%','uncertainty=20%','r')
% subplot(2,2,2);
% hold on
% plot(p1);
% hold on
% plot(p2);
% hold on
% plot(p3);
%
legend('M1,uncertainty=10%','M2,uncertainty=10%','M3,uncertainty=10%','M1,uncertainty=20%','M2,uncertainty=
20%','M3,uncertainty=20%');
% subplot(2,2,3);
% hold on
% plot(u_1,'k');
% legend('uncertainty=10%','uncertainty=20%')
% subplot(2,2,4);
% hold on
% plot(u_2,'k');
% legend('uncertainty=10%','uncertainty=20%')

```

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

```

clear
clc
%% .....First Model.....
%% .....
[n11,d11,n21,d21]=Inputsys(1);
Gs11 = tf(n11,d11);
Ts=0.1;
Gd11 = c2d(Gs11,Ts,'zoh');
[num11,den11]=tfdata(Gd11,'v');
Gs21 = tf(n21,d21);
Gd21 = c2d(Gs21,Ts,'zoh');
[num21,den21]=tfdata(Gd21,'v');
sys_info = stepinfo(Gd11);
ts11 = sys_info.SettlingTime;
tr11=sys_info.RiseTime;
sys_info = stepinfo(Gd21);
ts21 = sys_info.SettlingTime;
tr21=sys_info.RiseTime;
t=1:Ts:30;
[g11,t11] = step(Gd11,t);
[g21,t21] = step(Gd21,t);
P11=floor(tr11/Ts);
P21=floor(tr21/Ts);
N11=floor( ts11/Ts);
N21=floor( ts21/Ts);
P1=max(P11,P21);
N=max(N11,N21);
M1=P1;
% .....Second.....
[n12,d12,n22,d22]=Inputsys(2);
Gs12 = tf(n12,d12);
Gd12 = c2d(Gs12,Ts,'zoh');
[num12,den12]=tfdata(Gd12,'v');
Gs22 = tf(n22,d22);
Gd22 = c2d(Gs22,Ts,'zoh');

```

```

[num22,den22]=tfdata(Gd22,'v');
sys_info = stepinfo(Gd12);
ts12 = sys_info.SettlingTime;
tr12=sys_info.RiseTime;
sys_info = stepinfo(Gd22);
ts22 = sys_info.SettlingTime;
tr22=sys_info.RiseTime;
[g12,t1] = step(Gd12,t);
[g22,t2] = step(Gd22,t);
P12=floor(tr12/Ts);
P22=floor(tr22/Ts);
N12=floor( ts12/Ts);
N22=floor( ts22/Ts);
P2=max(P12,P22);
N2=max(N12,N22);
M2=P2;
% ..... Third .....
[n13,d13,n23,d23]=Inputs3(3);
Gs13 = tf(n13,d13);
Gd13 = c2d(Gs13,Ts,'zoh');
[num13,den13]=tfdata(Gd13,'v');
Gs23 = tf(n23,d23);
Gd23 = c2d(Gs23,Ts,'zoh');
[num23,den23]=tfdata(Gd23,'v');
sys_info = stepinfo(Gd13);
ts13 = sys_info.SettlingTime;
tr13=sys_info.RiseTime;
sys_info = stepinfo(Gd23);
ts23 = sys_info.SettlingTime;
tr23=sys_info.RiseTime;
[g13,t13] = step(Gd13,t);
[g23,t23] = step(Gd23,t);
P13=floor(tr13/Ts);
P23=floor(tr23/Ts);
N13=floor( ts13/Ts);
N23=floor( ts23/Ts);
P3=max(P13,P23);
N3=max(N13,N23);
M3=P3;
P=max(M1,M2);
P=max(P,M3);
M=P;
P3=P; P2=P; P1=P;
M3=M; M2=M; M1=M;
% .....
% % ..... First Model .....
% % .....
% ..... Toeplitz Matrix .....
b11 = zeros(1,P1); b11(1,1)= g11(2);
a11 = g11(2:P1+1);
G11 = toeplitz(a11,b11);
G11(:,M1) = G11(:,M1:P1)*ones(P1-M1+1,1);
G11 = G11(:,1:M1);
% .....
b21 = zeros(1,P1); b21(1,1)= g21(2);
a21 = g21(2:P1+1);

```

```

G21 = toeplitz(a21,b21);
G21(:,M1) = G21(:,M1:P1)*ones(P1-M1+1,1);
G21 = G21(:,1:M1);
G1=[G11 G21];
% .....
% A1~=-1-2.564z^-1+2.2365z^-2-0.6725z^-3
% According to the discrete transfer function, below parameters have been
% defined
na=3;
nb1=1; nb2=1;
nb=nb1;
d=0;
N11=d+1;
N21=d+P1;
% .....
a1_=[1 -2.564 2.2365 -0.6725];
% .....
b11_=num11(2:end);
b21_=num21(2:end);
C=1; % because of using white noise
f1=zeros(P1+d,na+1);
f1(1,1:3)=-1*a1_(2:4);
for j=1:P1+d-1
    for i=1:na
        f1(j+1,i)=f1(j,i+1)-f1(j,1)*a1_(i+1);
    end
end
F1=f1(N11:N21,1:na);
% .....
E11=zeros(P1);
E11(:,1)=1;
for j=1:P1-1
    E11(j+1:P1,j+1)=f1(j,1);
end
B11=zeros(P1,P1+nb);
for k=1:P1
    B11(k,k:k+1)=b11_;
end
m11_=E11*B11;
M11_=zeros(P1,nb+d);
for k=1:P1
    M11_(k,:)=m11_(k,k+1);
end
% .....
E21=zeros(P1);
E21(:,1)=1;
for j=1:P1-1
    E21(j+1:P1,j+1)=f1(j,1);
end
B21=zeros(P1,P1+nb);
for k=1:P1
    B21(k,k:k+1)=b21_;
end
m21_=E21*B21;
M21_=zeros(P1,nb+d);
for k=1:P1

```

```

M21_(k,:)=m21_(k,k+1);
end
M1_=[M11_ M21_];
%.....
gamma =1;
gain_DC11=(num11(1)+num11(2)+num11(3))/(den11(1)+den11(2)+den11(3));
gain_DC21=(num21(1)+num21(2)+num21(3))/(den21(1)+den21(2)+den21(3));
Q1 = eye(P1);
R11=((1.2)^2)*gamma*gain_DC11^2*eye(M1);
R21=gamma*gain_DC21^2*eye(M1);
R1=[R11 zeros(M1); zeros(M1) R21];
alpha=0.5;
Kgpc1=(G1'*Q1*G1+R1)\(G1'*Q1);
%% .....
%% .....Second
Model.....
%% .....
%.....Toeplitz Matrix.....
b12 = zeros(1,P2); b12(1,1)= g12(2);
a12 = g12(2:P2+1);
G12 = toeplitz(a12,b12);
G12(:,M2) = G12(:,M2:P2)*ones(P2-M2+1,1);
G12 = G12(:,1:M2);
% .....
b22 = zeros(1,P2); b22(1,1)= g22(2);
a22 = g22(2:P2+1);
G22 = toeplitz(a22,b22);
G22(:,M2) = G22(:,M2:P2)*ones(P2-M2+1,1);
G22 = G22(:,1:M2);
G2=[G12 G22];
% .....
%A2~=-1-2.411z^-1+1.9514z^-2-0.5404z^-3
% According to the discrete transfer function, below parameters have been
% defined
N12=d+1;
N22=d+P2;
% .....
a2_=[1 -2.474 2.0667 -0.5927];
% .....
b12_=num12(2:end);
b22_=num22(2:end);
C=1; % because of using white noise
f2=zeros(P2+d,na+1);
f2(1,1:3)=-1*a2_(2:4);
for j=1:P2+d-1
    for i=1:na
        f2(j+1,i)=f2(j,i+1)-f2(j,1)*a2_(i+1);
    end
end
F2=f2(N12:N22,1:na);
% .....
E12=zeros(P2);
E12(:,1)=1;
for j=1:P2-1

```



```

    E12(j+1:P2,j+1)=f2(j,1);
end
B12=zeros(P2,P2+nb);
for k=1:P2
    B12(k,k:k+1)=b12_;
end
m12_=E12*B12;
M12_=zeros(P2,nb+d);
for k=1:P2
    M12_(k,:)=m12_(k,k+1);
end
%.....
E22=zeros(P2);
E22(:,1)=1;
for j=1:P2-1
    E22(j+1:P2,j+1)=f2(j,1);
end
B22=zeros(P2,P2+nb);
for k=1:P2
    B22(k,k:k+1)=b22_;
end
m22_=E22*B22;
M22_=zeros(P2,nb+d);
for k=1:P2
    M22_(k,:)=m22_(k,k+1);
end
M2_=[M12_ M22_];
%.....
gamma2 =1;
gain_DC12=(num12(1)+num12(2)+num12(3))/(den12(1)+den12(2)+den12(3));
gain_DC22=(num22(1)+num22(2)+num22(3))/(den22(1)+den22(2)+den22(3));
Q2 = eye(P2);
R12 =((gain_DC22/gain_DC12)^2)*gamma2*gain_DC12^2*eye(M2);
R22=gamma2*gain_DC22^2*eye(M2);
R2=[R12 zeros(M2); zeros(M2) R22];
%alpha2=0.5;
Kgpc2=(G2'*Q2*G2+R2)/(G2'*Q2);
%%.....
%%.....Third
Model.....
%%.....
%.....Toeplitz Matrix.....
b13 = zeros(1,P3); b13(1,1)= g13(2);
a13 = g13(2:P3+1);
G13 = toeplitz(a13,b13);
G13(:,M3) = G13(:,M3:P3)*ones(P3-M3+1,1);
G13 = G13(:,1:M3);
%.....
b23 = zeros(1,P3); b23(1,1)= g23(2);
a23 = g23(2:P3+1);
G23 = toeplitz(a23,b23);
G23(:,M3) = G23(:,M3:P3)*ones(P3-M3+1,1);
G23 = G23(:,1:M3);
G3=[G13 G23];

```

```

% .....
% A3~=-1-2.725z^-1+2.5357z^-2-0.8107z^-3
% According to the discrete transfer function, below parameters have been
% defined
N13=d+1;
N23=d+P3;
% .....
a3_=[1 -2.411 1.9514 -0.5404];%[1 -2.725 2.5357 -0.8107];
% .....
b13_=num13(2:end);
b23_=num23(2:end);
C=1; % because of using white noise
f3=zeros(P3+d,na+1);
f3(1,1:3)=-1*a3_(2:4);
for j=1:P3+d-1
    for i=1:na
        f3(j+1,i)=f3(j,i+1)-f3(j,1)*a3_(i+1);
    end
end
F3=f3(N13:N23,1:na);
% .....
E13=zeros(P3);
E13(:,1)=1;
for j=1:P3-1
    E13(j+1:P3,j+1)=f3(j,1);
end
B13=zeros(P3,P3+nb);
for k=1:P3
    B13(k,k:k+1)=b13_;
end
m13_=E13*B13;
M13_=zeros(P3,nb+d);
for k=1:P3
    M13_(k,:)=m13_(k,k+1);
end
% .....
E23=zeros(P3);
E23(:,1)=1;
for j=1:P3-1
    E23(j+1:P3,j+1)=f3(j,1);
end
B23=zeros(P3,P3+nb);
for k=1:P3
    B23(k,k:k+1)=b23_;
end
m23_=E23*B23;
M23_=zeros(P3,nb+d);
for k=1:P3
    M23_(k,:)=m23_(k,k+1);
end
M3_=[M13_ M23_];
% .....
gamma3 =1;
gain_DC13=(num13(1)+num13(2)+num13(3))/(den13(1)+den13(2)+den13(3));
gain_DC23=(num23(1)+num23(2)+num23(3))/(den23(1)+den23(2)+den23(3));
Q3 = eye(P3);

```



```

u_2=[];
ym1=[];
y1=0;
Y_d1=zeros(P1,length(t));
Y_past1=zeros(P1,length(t));
Y_m1=zeros(P1,length(t));
D1=zeros(P1,length(t));
E1=zeros(P1,length(t));
dU11=zeros(M1,length(t));
dU21=zeros(M1,length(t));
dU1=[dU11;dU21];
U11=zeros(M1,length(t));
U21=zeros(M1,length(t));
Y1_=zeros(na,length(t));
dU001=dU1;
dU001(1,1)=0.001;
dU001(P1+1,1)=0.001;
%.....second.....
dU12_=zeros(nb+d,length(t));
dU22_=zeros(nb+d,length(t));
dU2_=[dU12_;dU22_];
d2=zeros(1,length(t));
%.....
ym2=[];
y2=0;
Y_d2=zeros(P2,length(t));
Y_past2=zeros(P2,length(t));
Y_m2=zeros(P2,length(t));
D2=zeros(P2,length(t));
E2=zeros(P2,length(t));
dU12=zeros(M2,length(t));
dU22=zeros(M2,length(t));
dU2=[dU12;dU22];
U12=zeros(M2,length(t));
U22=zeros(M2,length(t));
Y2_=zeros(na,length(t));
dU002=dU2;
dU002(1,1)=0.001;
dU002(P2+1,1)=0.001;
%.....Third.....
dU13_=zeros(nb+d,length(t));
dU23_=zeros(nb+d,length(t));
dU3_=[dU13_;dU23_];
d3=zeros(1,length(t));
%.....
ym3=[];
y3=0;
Y_d3=zeros(P3,length(t));
Y_past3=zeros(P3,length(t));
Y_m3=zeros(P3,length(t));
D3=zeros(P3,length(t));
E3=zeros(P3,length(t));
dU13=zeros(M3,length(t));
dU23=zeros(M3,length(t));
dU3=[dU13;dU23];
U13=zeros(M3,length(t));

```

```

U23=zeros(M3,length(t));
Y3_=zeros(na,length(t));
dU003=dU3;
dU003(1,1)=0.001;
dU003(P3+1,1)=0.001;
dU01=zeros(M,length(t));
dU02=zeros(M,length(t));
dU=[dU01; dU02];
% .....
% .....
p=zeros(3,length(t));
p(1:3,2)=[0.4 0.2 0.4]';
p_=zeros(3,length(t));
p__=zeros(3,length(t));
e=zeros(2,length(t),3);
p1=p(1,2);
p2=p(2,2);
p3=p(3,2);

for i=1:length(t)-1

for j=1:P1
    Y_d1(j,i+1)=(alpha^j)*y1+(1-(alpha)^j)*(r(i+1)-441.2); % Programmed
end

Y_past1(:,i+1)=M1_*dU1_(:,i+1)+F1*Y1_(:,i+1);
D1(:,i+1)=d1(i+1)*ones(P1,1);

E1(:,i+1)=Y_d1(:,i+1)-Y_past1(:,i+1)-D1(:,i+1);

H1 = 2*(G1'*Q1*G1+R1);
f1 = -(2*E1(:,i+1))*Q1*G1';
ub11=up1*ones(P1,1)-(U11(1,i)+100);
ub21=up2*ones(P1,1)-(U21(1,i)+100);
ub1=[ub11; ub21];
lb11=low1*ones(P1,1)-(U11(1,i)+100);
lb21=low2*ones(P1,1)-(U21(1,i)+100);
lb1=[lb11; lb21];
%interior-point-convex
%'trust-region-reflective'
opts1 = optimoptions('quadprog','Algorithm','interior-point-convex','Display','off');
dU1(:,i+1)= quadprog(H1,f1,[],[],[],[],lb1,ub1,dU001(:,i),opts1);
dU001(:,i+1)=dU1(:,i+1);
dU11(:,i+1)=dU1(1:M1,i+1);
dU21(:,i+1)=dU1(M1+1:2*M1,i+1);
% U11(1,i+1)=dU11(1,i+1)+U11(1,i);
% U21(1,i+1)=dU21(1,i+1)+U21(1,i);
dU1(:,i+1)=[dU11(:,i+1);dU21(:,i+1)];

% .....
u11=U11(1,i)+100;
u21=U21(1,i)+100;

```

```

%% .....
%% .....Second
Model.....
%% .....
% .....

for j=1:P2
    Y_d2(j,i+1)=(alpha^j)*y2+(1-(alpha)^j)*(r(i+1)-443.5); % Programmed
end

Y_past2(:,i+1)=M2_*dU2(:,i+1)+F2*Y2(:,i+1);
D2(:,i+1)=d2(i+1)*ones(P2,1);

E2(:,i+1)=Y_d2(:,i+1)-Y_past2(:,i+1)-D2(:,i+1);

H2 = 2*(G2'*Q2*G2+R2);
f2 = -(2*E2(:,i+1)'*Q2*G2)';
ub12=up1*ones(P2,1)-(U12(1,i)+100);
ub22=up2*ones(P2,1)-(U22(1,i)+97);
ub2=[ub12; ub22];
lb12=low1*ones(P2,1)-(U12(1,i)+100);
lb22=low2*ones(P2,1)-(U22(1,i)+97);
lb2=[lb12; lb22];
opts2 = optimoptions('quadprog','Algorithm','interior-point-convex','Display','off');
dU2(:,i+1)= quadprog(H2,f2,[],[],[],lb2,ub2,dU002(:,i),opts2);
dU002(:,i+1)=dU2(:,i+1);
dU12(:,i+1)=dU2(1:M2,i+1);
dU22(:,i+1)=dU2(M2+1:2*M2,i+1);
% U12(1,i+1)=dU12(1,i+1)+U12(1,i);
% U22(1,i+1)=dU22(1,i+1)+U22(1,i);
dU2(:,i+1)=[dU12(:,i+1);dU22(:,i+1)];

% .....
u12=U12(1,i)+100;
u22=U22(1,i)+97;
% .....
%% .....
%% .....Third
Model.....
%% .....
% .....

for j=1:P3
    Y_d3(j,i+1)=(alpha^j)*y3+(1-(alpha)^j)*(r(i+1)-445.3); % Programmed
end

Y_past3(:,i+1)=M3_*dU3(:,i+1)+F3*Y3(:,i+1);

```

```

D3(:,i+1)=d3(i+1)*ones(P3,1);

E3(:,i+1)=Y_d3(:,i+1)-Y_past3(:,i+1)-D3(:,i+1);

H3 = 2*(G3*Q3*G3+R3);
f3 = -(2*E3(:,i+1)*Q3*G3)';
ub13=up1*ones(P3,1)-(U13(1,i)+103);
ub23=up2*ones(P3,1)-(U23(1,i)+97);
ub3=[ub13; ub23];
lb13=low1*ones(P3,1)-(U13(1,i)+103);
lb23=low2*ones(P3,1)-(U23(1,i)+97);
lb3=[lb13; lb23];
opts3 = optimoptions('quadprog','Algorithm','interior-point-convex','Display','off');
dU3(:,i+1)= quadprog(H3,f3,[],[],[],lb3,ub3,dU003(:,i),opts3);
dU003(:,i+1)=dU3(:,i+1);
dU13(:,i+1)=dU3(1:M3,i+1);
dU23(:,i+1)=dU3(M3+1:2*M3,i+1);
% U13(1,i+1)=dU13(1,i+1)+U13(1,i);
% U23(1,i+1)=dU23(1,i+1)+U23(1,i);
dU3(:,i+1)=[dU13(:,i+1);dU23(:,i+1)];
% .....
% we should give models U which has obtained with 3 models with weights
dU=p(1,i+1)*dU1+p(2,i+1)*dU2+p(3,i+1)*dU3;
dU01(:,i+1)=dU(1:M,i+1);
dU02(:,i+1)=dU(M+1:2*M,i+1);
% .....

Y_m1(:,i+1)=G1*dU(:,i+1)+Y_past1(:,i+1);

dU11_(2:nb+d,i+2) = dU11_(1:nb+d-1,i+1);
dU11_(1,i+2)=dU01(1,i+1);
dU21_(2:nb+d,i+2) = dU21_(1:nb+d-1,i+1);
dU21_(1,i+2)=dU02(1,i+1);
dU1_(:,i+2)=[dU11_(:,i+2);dU21_(:,i+2)];
Y1_(2:na,i+2)=Y1_(1:na-1,i+1);
Y1_(1,i+2)=Y_m1(1,i+1);
% .....
Y_m2(:,i+1)=G2*dU(:,i+1)+Y_past2(:,i+1);

dU12_(2:nb+d,i+2) = dU12_(1:nb+d-1,i+1);
dU12_(1,i+2)=dU01(1,i+1);
dU22_(2:nb+d,i+2) = dU22_(1:nb+d-1,i+1);
dU22_(1,i+2)=dU02(1,i+1);
dU2_(:,i+2)=[dU12_(:,i+2);dU22_(:,i+2)];
Y2_(2:na,i+2)=Y2_(1:na-1,i+1);
Y2_(1,i+2)=Y_m2(1,i+1);
% .....

Y_m3(:,i+1)=G3*dU(:,i+1)+Y_past3(:,i+1);

dU13_(2:nb+d,i+2) = dU13_(1:nb+d-1,i+1);
dU13_(1,i+2)=dU01(1,i+1);
dU23_(2:nb+d,i+2) = dU23_(1:nb+d-1,i+1);
dU23_(1,i+2)=dU02(1,i+1);

```

```

dU3_(:,i+2)=[dU13_(:,i+2);dU23_(:,i+2)];
Y3_(2:na,i+2)=Y3_(1:na-1,i+1);
Y3_(1,i+2)=Y_m3(1,i+1);

% .....
u13=U13(1,i)+103;
u23=U23(1,i)+97;
% % .....
u1=p(1,i+1)*u11+p(2,i+1)*u12+p(3,i+1)*u13+dU01(1,i+1);
u2=p(1,i+1)*u21+p(2,i+1)*u22+p(3,i+1)*u23+dU02(1,i+1);
U13(1,i+1)=u1-103;
U23(1,i+1)=u2-97;
U12(1,i+1)=u1-100;
U22(1,i+1)=u2-97;
U11(1,i+1)=u1-100;
U21(1,i+1)=u2-100;
% .....
sim('Model')
% .....
d1(i+2)=y(end)-Y_m1(1,i+1)-441.2;
d2(i+2)=y(end)-Y_m2(1,i+1)-443.5;
d3(i+2)=y(end)-Y_m3(1,i+1)-445.3;
% .....
e(2:end,i+2,1)=e(1:end-1,i+1,1);
e(1,i+2,1)=abs(d1(i+2));
e(2:end,i+2,2)=e(1:end-1,i+1,2);
e(1,i+2,2)=abs(d2(i+2));
e(2:end,i+2,3)=e(1:end-1,i+1,3);
e(1,i+2,3)=abs(d3(i+2));
% .....
sum=0;
for k=1:3
    sum=1/e(1,i+2,k)+sum;
end
w=zeros(3,1);
for j=1:3
    w(j)=(1/e(1,i+2,j))/sum;
end
% for j=1:3
%     p(j,i+2)=w(j);
% end
[max1,ind1]=max(w);
for j=1:3
    if j==ind1
        w(j)=-200;
    end
end
[max2,ind2]=max(w);
% res=[0.5 0; 0 0.1];
% sum2=0;
% for j=1:3
%     sum2=exp(e(:,i+2,j))*res*e(:,i+2,j))*p(j,i+1)+sum2;
% end
% p_(ind1,i+2)=(exp(e(:,i+2,ind1))*res*e(:,i+2,ind1))*p(j,i+1))/sum2;
% p_(ind2,i+2)=(exp(e(:,i+2,ind2))*res*e(:,i+2,ind2))*p(j,i+1))/sum2;
% delta=0.01;

```



```

% if p_(ind1,i+2)>=delta
%   p__(ind1,i+2)=p_(ind1,i+2);
% else
%   p__(ind1,i+2)=delta;
% end
% if p_(ind2,i+2)>=delta
%   p__(ind2,i+2)=p_(ind2,i+2);
% else
%   p__(ind2,i+2)=delta;
% end
% sum3=0;
% for j=1:3
%   sum3=sum3+p__(j,i+2);
% end
% for j=1:3
%   p(j,i+2)=p__(j,i+2)/sum3;
% end
p(ind1,i+2)=max1;
p(ind2,i+2)=max2;
for j=1:3
    if j~=ind1 && j~=ind2
        p(j,i+2)=0;
    end
end
sum3=0;
for j=1:3
    sum3=sum3+p(j,i+2);
end
for j=1:3
    p(j,i+2)=p(j,i+2)/sum3;
end
% .....
y1=y(end)-441.2;%+dist(i,1); % nonlinear
y2=y(end)-443.5;%+dist(i,1); % nonlinear
y3=y(end)-445.3;%+dist(i,1); % nonlinear
% .....
% .....
% y1=[y1; y+441.2];
% y1=[y1; y+445.3];
ynl=[ynl; y(end)];
% .....
ym1=[ym1; Y_m1(1,i)];
ym2=[ym1; Y_m2(1,i)];
ym3=[ym1; Y_m3(1,i)];
u_1=[u_1; u1];
u_2=[u_2; u2];
x01=x1(end);
x02=x2(end);
p1=[p1; p(1,i+2)];
p2=[p2; p(2,i+2)];
p3=[p3; p(3,i+2)];
x_1=[x_1; x1(end)];
end

figure(1);
subplot(3,2,1);

```

```

plot(ynl,'b');
hold on
% .....
plot(r,'r');
% plot(r+445.3,'r');
% plot(r+436.8,'r');
% .....
grid on
legend('y','r');
title('Response of the nonlinear system');
xlabel('sample');
subplot(3,2,2);
% .....
% plot(y1-441.2,'b');
% plot(y1-445.3,'b');
plot(ynl,'b');
% .....
hold on
plot(ym1+441.2,'m');
grid on
hold on
plot(ym2+443.5,'c');
hold on
plot(ym3+445.3,'r');
xlabel('sample');
title('Yms with bias and Yp');
legend('YPlant','YModel1','YModel2','YModel3');
subplot(3,2,3);
plot(u_1,'b');
grid on
xlabel('sample');
title('Control law for input 1');
subplot(3,2,4);
plot(u_2,'b');
grid on
xlabel('sample');
title('Control law for input 2');
subplot(3,2,6);
plot(p1,'b');
hold on
plot(p2,'m');
hold on
plot(p3,'c');
legend('Model1','Model2','Model3');
grid on
xlabel('sample');
title('Weights of Models');
subplot(3,2,5);
plot(x_1)
grid on
xlabel('sample');
title('state one');

```

شبیه سازی صورت گرفته در سیمولینک متلب مشابه قبل می باشد.