دانشکده مهندسی مکانیک

مبانی طراحی کنترل اتوماتیک
دکتر آریا الستی

پروژه درس

یاشار زعفری حقی
۹۹۱۰۶۲۰۹
عرفان رادفر
۹۹۱۰۹۶۰۳

۱۱ بهمن ۱۴۰۲

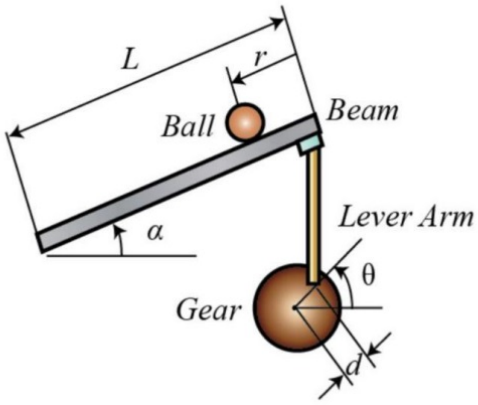
مبانی طراحی کنترل اتوماتیک

پروژه درس

یاشار زعفری حقی - ۹۹۱۰۶۲۰۹
عرفان رادفر - ۹۹۱۰۹۶۰۳

تعریف مسئله

سیستم Ball and Beam از معروف ترین و ساده‌ترین سیستم‌های کنترل است. این سیستم شامل یک تیر بلند است که قابلیت حرکت توپ داخل آن را دارد. هدف کنترلی در این سیستم، کنترل مکان توپ دقیقا در وسط تیر است. به این منظور یک سنسور التراسونیک برای تشخیص مکان و سرعت توپ در هر لحظه و یک سروو موتور در وسط یا اطراف تیر برای تولید حرکت دورانی در تیر و کنترل مکان توپ تعبیه شده است.



فایل شبیه‌سازی شده‌ی این سیستم با عنوان modlBB۱۵ موجود می‌باشد که می‌توان از طریق آن، رابطه میان زاویه‌ی موتور (θ) و موقعیت تو بر روی تیر (r) را استخراج کرد.
می‌خواهیم زاویه θ را با یک موتور و گیربکس کاهنده با ضریب ۵ کنترل کنیم. تابع تبدیل موتور به صورت زیر می‌باشد:

$$\frac{\theta}{V} = \frac{0.0274}{0.003228s^2 + 0.003508s}$$

رابطه‌ی ولتاژ با مکان مطلوب بصورت زیر است:

$$R = 2V$$

توضیحات

- در بخش‌های ۱، ۲، ۵، و ۶ با استفاده از تابع تبدیل بدست آمده برای سیستم، کنترلر را طراحی کنید و آنرا با مدل شبیه سازی شده ارزیابی نمایید.
- همه‌ی کنترلرهای طراحی شده باید علّی باشند. (رسته ی صورت کمتر از رسته ی مخرج باشد.)
- دقیقا قبل از موتور یک حد اشباع با حد ۲۰ قرار دهید.
- فایل‌های متلب را با ورژن ۲۰۲۲ یا کمتر و با فرمت slx ذخیره کنید. (نام فایل مربوط به هر قسمت در گزارش ذکر شود.)

- گزارش پروژه باید در قالب یک فایل پی دی اف و بصورت تایپ شده باشد. گزارش باید کامل، گویا و شامل توضیحات، نتایج، نمودارها و تحلیل‌های انجام شده باشد. تمامی تصاویر و نمودارهای استفاده شده در گزارش و تمامی فایل‌های متلب استفاده شده باید به فایل پی دی اف ضمیمه شوند.
- فایل فشرده‌ی پروژه را در زمان مقرر در سامانه درس‌افزار شریف بارگذاری نمایید. (تحویل با تاخیر پذیرفته نیست.)
- پروژه بصورت گروههای دو نفره تعریف شدهاست و برای ارائه ی پروژه، هر دو نفر باید حاضر باشند.
- علاوه بر فایل پی دی اف پروژه، یک فایل پاورپوینت برای ارائه نیز آماده گردد.

خواسته‌ها

- فراجهش کمتر از ۲۰ درصد
- زمان نشست کمتر از ۸ ثانیه. معیار زمان نشست ما مطابق متلب و ۲ درصد می‌باشد.

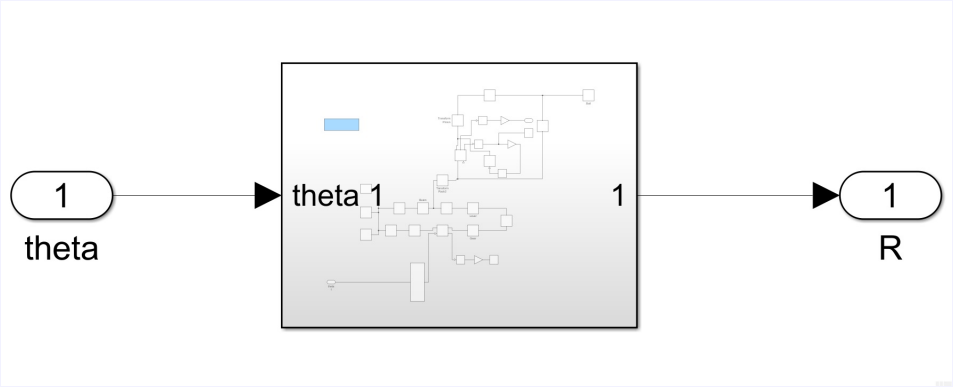
لطفاً در هر مرحله فقط فولدر مربوط به آن سوال فعال باشد و بقیه غیر فعال باشند.

۱. با استفاده از مدل شبیه‌سازی شده و بهره‌گیری از روش‌های شناسایی سیستم (System Identification) تابع تبدیل $\frac{R}{\theta}$ را بدست آورید.

پاسخ

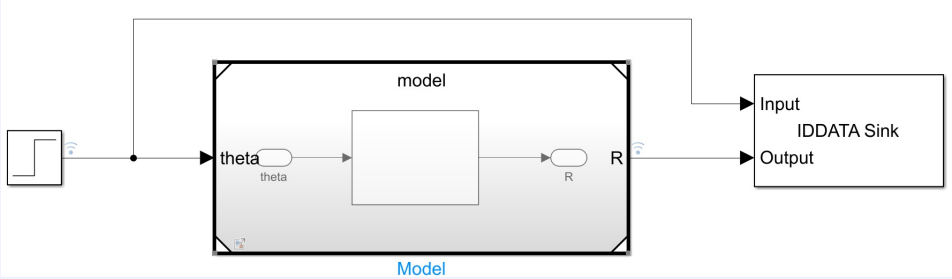
فایل‌های مربوط به این بخش در فولدر q۱ قرار دارند.

ابتدا مدل مکانیکی داده شده برای سیستم را بررسی می‌کنیم. برای راحتی، یک مدل رفرنس بدون ورودی پله و نمودار خروجی، به نام model.slx از مدل داده شده ایجاد می‌کنیم:



شکل ۱: مدل رفرنس

حال برای بهره‌گیری از روش‌های شناسایی سیستم، نیازمند دیتایی هستیم که رفتار سیستم را به ازای یک ورودی مشخص نشان دهد. بدین منظور با شبیه‌سازی در سیمولینک، فایل sys_ident، سیگنال ورودی پله به اندازه ۰.۵ و بدون تأخیر را به سیستم رفرنس اعمال میکنیم، و با استفاده از بلوک IDDATA دیتای مورد نیاز برای را ایجاد می‌کنیم، سپس با استفاده از Callback Function InitFcn دیتای ایجاد شده را با نام rawdata ذخیره می‌کنیم:

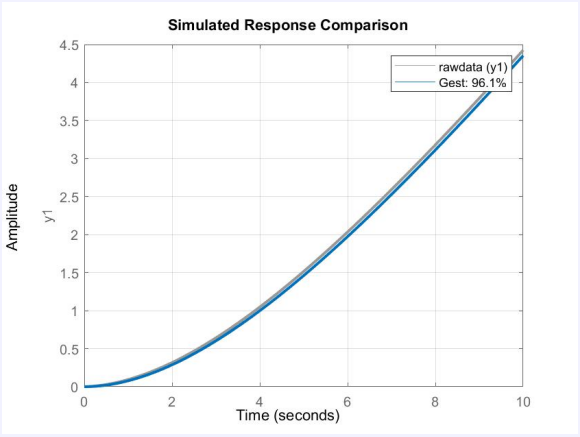


شکل ۲: ایجاد دیتا برای شناسایی سیستم

حال با استفاده از دیتای ایجاد شده، سیستم را شناسایی می‌کنیم. بدین منظور، فایل لایو اسکریپت OurSystemIdent را ایجاد کردیم. نتایج به صورت زیر می‌باشند:

- بر مبنای تابع تبدیل:

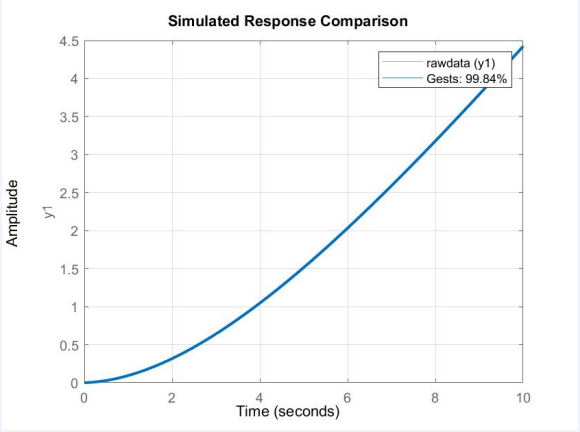
$$G_{est} = \frac{0.335}{s^2 + 0.2381s + 9.649e-07}$$



شکل ۳: شناسایی بر مبنای تابع تبدیل

- بر مبنای فضای حالت:

$$G_{ests} = \frac{0.03613s + 0.335}{s^2 + 0.2381s + 1.229e-15}$$



شکل ۴: شناسایی بر مبنای فضای حالت

با توجه به اینکه درصد شباهت تخمین بر مبنای فضای حالت بیشتر است، آن را مبنای طراحی کنترلرهای خود قرار می‌دهیم و با صرف نظر از ترم بسیار کوچک در مخرج، نتیجه‌ی شناسایی سیستم و در نتیجه تخمین سیستم به صورت زیر می‌باشد:

$$G_{ests} = \frac{0.03613s + 0.335}{s^2 + 0.2381s}$$

۲. با استفاده از جعبه ابزار SISO کنترلری از خانواده‌ی PID طراحی کنید بگونه‌ای که شرایط فوق حاصل گردد.
(ممکن است صرفا با یک کنترلر PID نتوان به خواسته‌های مسئله رسید. در اینصورت می‌توانید یک کنترلر کمکی در مدار

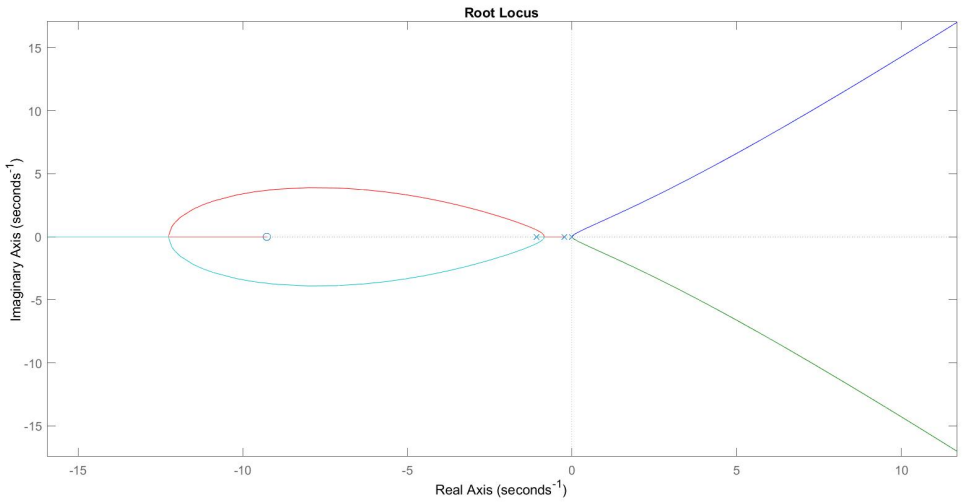
قرار دهید (به هر روش دلخواه مانند lead ، IMC و ...) که پایداری سیستم افزایش یابد و سپس کنترلر PID را طراحی کنید. از این کنترلر کمکی برای افزایش پایداری می‌توانید در قسمت‌های بعد نیز در صورت لزوم استفاده کنید.)

پاسخ

فایل‌های مربوط به این بخش در فولدر q۲ قرار دارند.

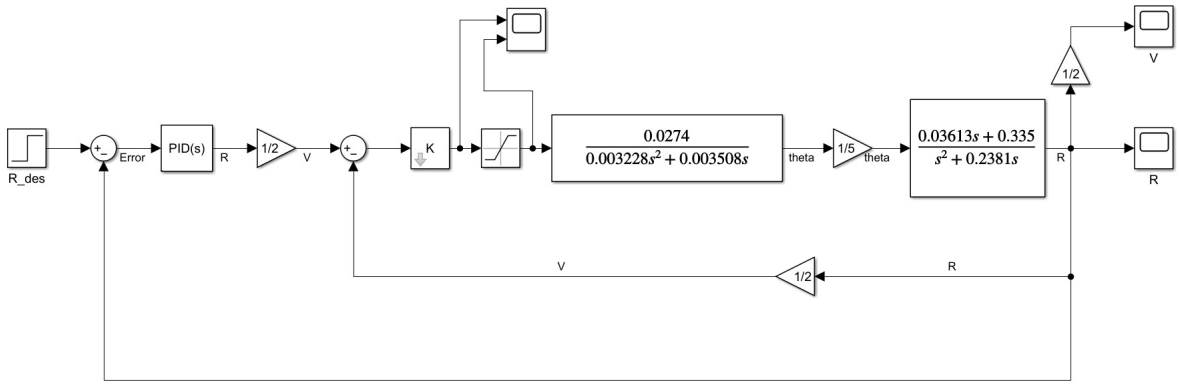
در این بخش ابتدا مکان هندسی ریشه‌های مدار باز را بررسی می‌کنیم:

$$G_{OLTF} = G_{motor} \times G_{plant} \times \frac{1}{2} \times \frac{1}{5} = \frac{0.00099s + 0.009179}{0.03228s^4 + 0.04277s^3 + 0.008353s^2}$$



شکل ۵: مکان هندسی ریشه‌ها

همانطور که مشخص می‌باشد، دو شاخه ناپایدار داریم. حال با توجه به اینکه می‌خواهیم کنترلر PID طراحی کنیم، با اضافه شدن قطب مبدأ انتگرال‌گیر شاخه‌های ناپایدار بیشتر به سمت راست کشیده می‌شوند و عملاً با کنترلر سیستم را ناپایدارتر می‌کنیم. بدین منظور نیازمندیم تا حدی پایداری سیستم را با یک کنترلر کمکی ارتقا بدهیم. پس از ضریب کالیبراسیون ولتاژ و مکان استفاده کرده و یک فیدبک دیگر از سیستم گرفته و یک مدار داخلی تشکیل می‌دهیم و در آن کنترلر K را طراحی می‌کنیم. تا ابتدا حلقه‌ی داخلی پایدار شود و سپس با کنترلر PID حلقه بیرونی را تنظیم می‌کنیم تا به خواسته‌های طراحی برسیم. بدین منظور ابتدا شماتیک سیستم مدار بسته با مدار داخلی به صورت شکل زیر می‌باشد:



شکل ۶: شماتیک سیستم مدار بسته

حال پایدارساز داخلی را طراحی می‌کنیم. بدین منظور از روش IMC و قضیه یولا کوچرا استفاده می‌کنیم. ابتدا پلنت حلقه داخلی را بررسی می‌کنیم:

$$G = G_{motor} \times \frac{1}{5} \times G_{plant} \times \frac{1}{2} = \frac{0.00099s + 0.009179}{0.03228s^4 + 0.04277s^3 + 0.008353s^2}$$

توجه داشته باشید که در این روش فیدبک واحد داریم و بدین منظور gain موجود در فیدبک را به فیدفوروارد انتقال می‌دهیم. و از طرفی چون صرفاً با این کنترلر، می‌خواهیم پایداری را بدست بیاوریم، نیازی به اعمال gain در بیرون حلقه پس از انتقال آن نیست و این انتقال gain صرفاً برای طراحی کنترلر می‌باشد. با توجه به پایدار بودن پلنت، در قضیه بزو و یولا کوچرا داریم:

$$X = 0,\ Y = 1,\ M = 1,\ N = G \rightarrow K(s) = \frac{Q(s)}{1 - G(s)Q(s)}$$

برای تبعیت از فرمان، می‌توانیم از معکوس G به عنوان Q استفاده کنیم. ولی به دلیل اکیداً سره بودن نمی‌توانیم مستقیماً از معکوس آن استفاده کنیم و ناچاریم از معکوس تقریبی آن استفاده کنیم:

$$Q(s) = b\tilde{G}^{-1}(s) = \frac{b}{G(s)(s+a)^k}$$

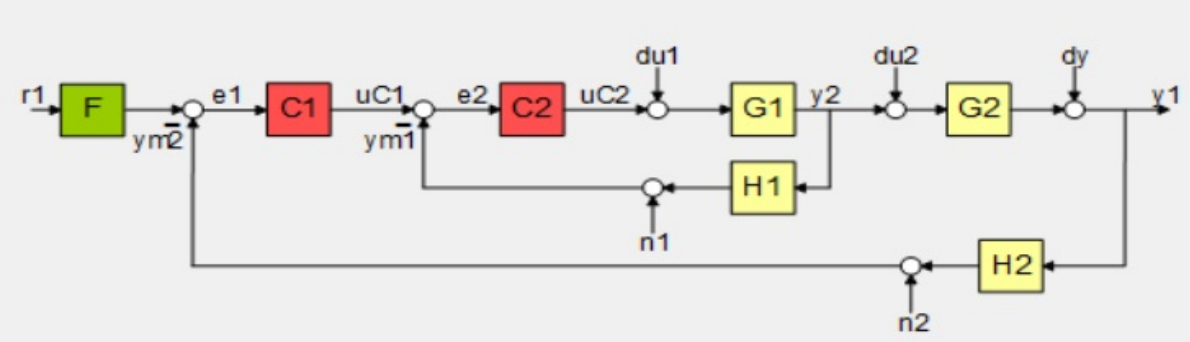
با توجه به اینکه رسته نسبی G، ۳ می‌باشد، k حداقل باید ۳ باشد. ما طی چندبار آزمایش و خطا، تصمیم گرفتیم که k=۵ باشد. همچنین a=۴ قرار می‌دهیم. همچنین b را در نهایت با در نظر گرفتن اینکه بهره‌ی صفر مسیر فیدفوروارد ۱ باشد، تعیین کردیم. در نتیجه داریم:

$$Q(s) = \frac{b(0.03228s^4 + 0.04277s^3 + 0.008353s^2)}{(s+4)^5\ (0.00099s + 0.009179)} \rightarrow$$

$$\rightarrow K = \frac{33390s^2(s + 1.087)(s + 0.2381)}{(s + 16.53)(s + 2.02)(s^2 + 3.914s + 5.928)(s^2 + 6.808s + 47.98)}$$

حال از این کنترلر داخلی، در تمامی بخش‌ها استفاده کرده، و کنترلر PID را طراحی می‌کنیم تا به خواسته‌های موردنظر برسیم. کد اسکریپت موارد گفته شده با نام test_IMC در فولدر موجود می‌باشد.

حال با قرار دادن این کنترلر در حلقه داخلی، با استفاده از جعبه ابزار SISO سعی می‌کنیم کنترلر PID را طراحی کنیم که خواسته‌های مسئله را برآورده کند. کنترلر PID را می‌توانیم با اضافه کردن صفر مختلط، قطب حقیقی و انتگرال‌گیر در کنترلر ایجاد کرد. سپس با جابجا کردن این نقاط در نمودار مکان هندسی آنها را تنظیم کرد. ابتدا ساختار سیستم را به صورت زیر انتخاب می‌کنیم که در آن داریم: (کد q۲ موجود در فولدر صرفاً برای تعریف توابع تبدیل و اعمال آن در جعبه‌ابزار می‌باشد.)



شکل ۷: ساختار

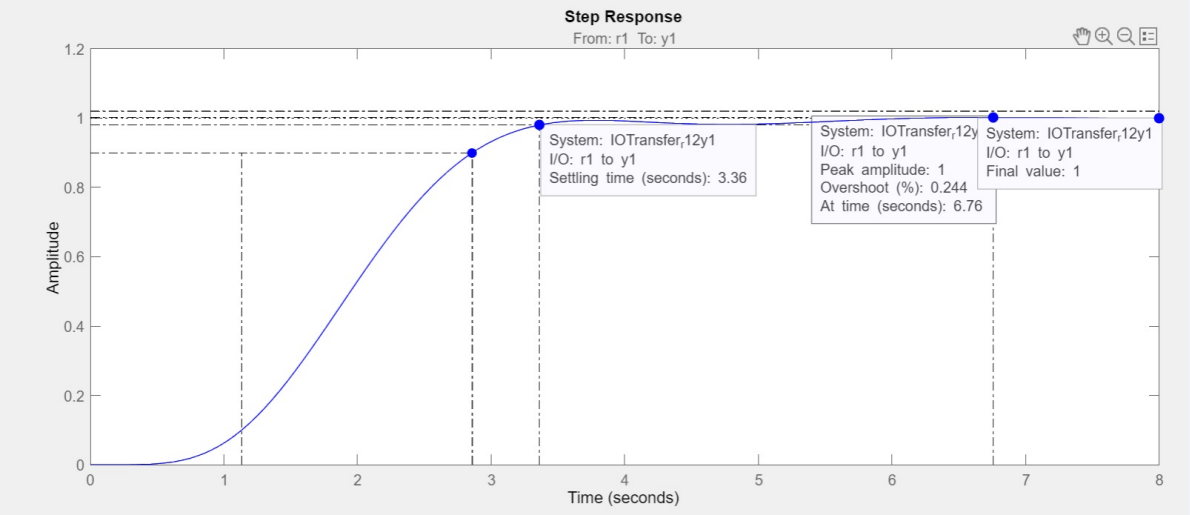
$$G_1 = G_{motor} \times G_{plant} = \frac{0.06134s + 0.5687}{s^2 + 1.325s + 0.2588},\ F = 0.5,\ H_1 = 0.5,\ H_2 = 0.5,\ G_2 = 1,$$

$$C_2 = \frac{33390.0\,s^2 + 44240.0\,s + 8640.0}{s^6 + 29.27\,s^5 + 312.8\,s^4 + 2080.0\,s^3 + 7206.0\,s^2 + 12890.0\,s + 9495.0}$$

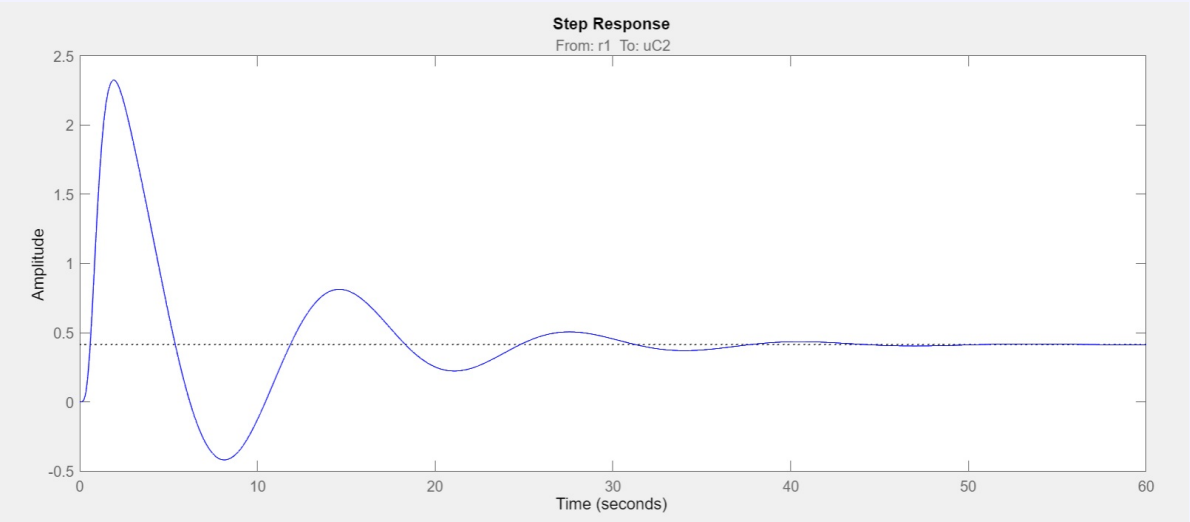
توجه شود که جعبه‌ابزار SISO خود به خود متوجه حذف صفر و قطب پلنت داخلی و کنترلر داخلی نمی‌شود و به همین دلیل خودمان آنها را حذف کردیم تا نتایج به درستی در جعبه‌ابزار ظاهر شود. با تنظیم، به کنترلر زیر رسیدیم:

$$C_1 = \frac{0.0074813(s^2 + 40s + 401)}{s(s+3)} \rightarrow C_1 = K_p + K_i\frac{1}{s} + K_d\frac{s}{T_f s + 1}$$

$$Kp = -0.234,\ Ki = 1,\ Kd = 0.0804,\ Tf = 0.333 \rightarrow N = 3$$

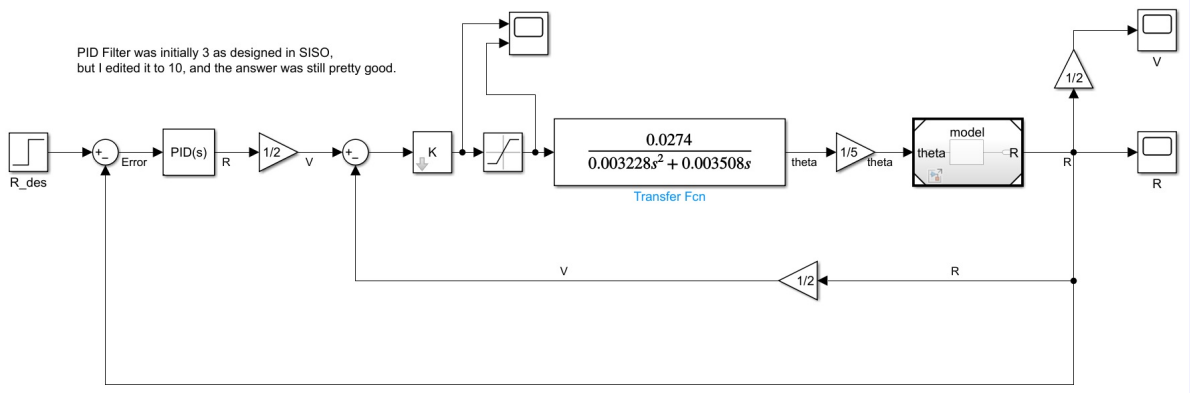


شکل ۸: پاسخ پله

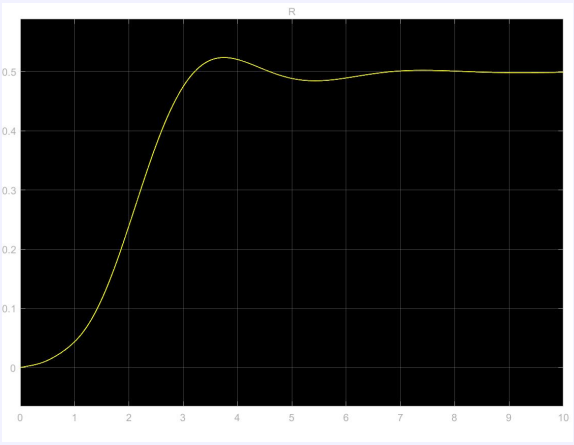


شکل ۹: سیگنال کنترلی عملگر

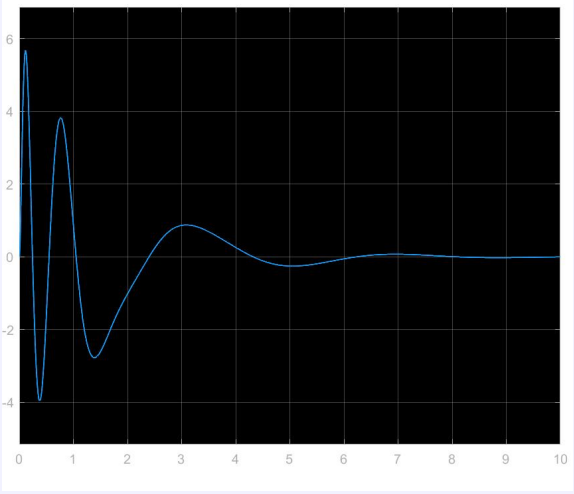
با توجه به اینکه ضریب فیلتر تاثیری چندانی در نتیجه ندارد، آنرا N=۱۰ قرار می‌دهیم. حال رفتار مدل واقعی را به ازای این کنترلر و ورودی دلخواه که نیم می‌باشد، بررسی می‌کنیم:



شکل ۱۰: بلوک دیاگرام سیستم



شکل ۱۱: پاسخ پله



شکل ۱۲: سیگنال کنترلی

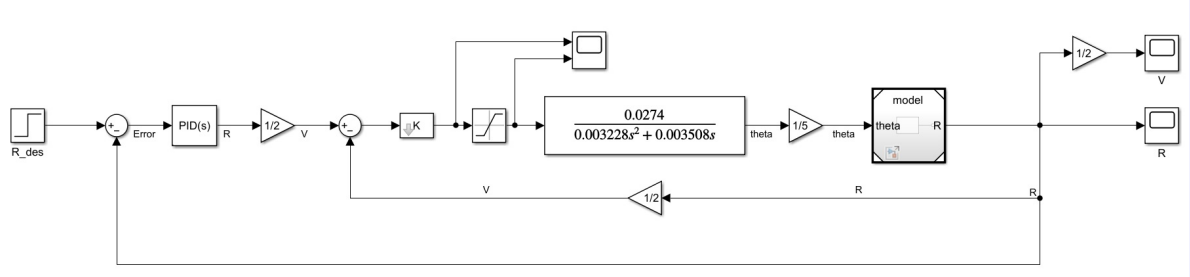
همانطور که مشاهده می‌شود پاسخ به نیم رسیده، و زمان نشست آن کمتر از ۸ ثانیه و اورشوت زیر ۲۰ درصد دارد و در نتیجه خواسته‌های مسئله را ارضاء می‌کند. از طرفی سیگنال کنترلی عملگر نیز، با توجه به حد اشباع ۲۰، دچار اشباع نمی‌شود و در نتیجه از این بابت هم نیاز به سیستم آنتی ویندآپ نداریم. فایل سیمولینک مربوطه با نام filter_N_۱۰۰ در فولدر موجود می‌باشد.

۳. به کمک ابزار PID-Tuner متلب، کنترلری از خانواده‌ی PID طراحی کنید بگونه‌ای که شرایط فوق حاصل گردد.

پاسخ

فایل‌های مربوط به این بخش در فولدر q۳ قرار دارند.

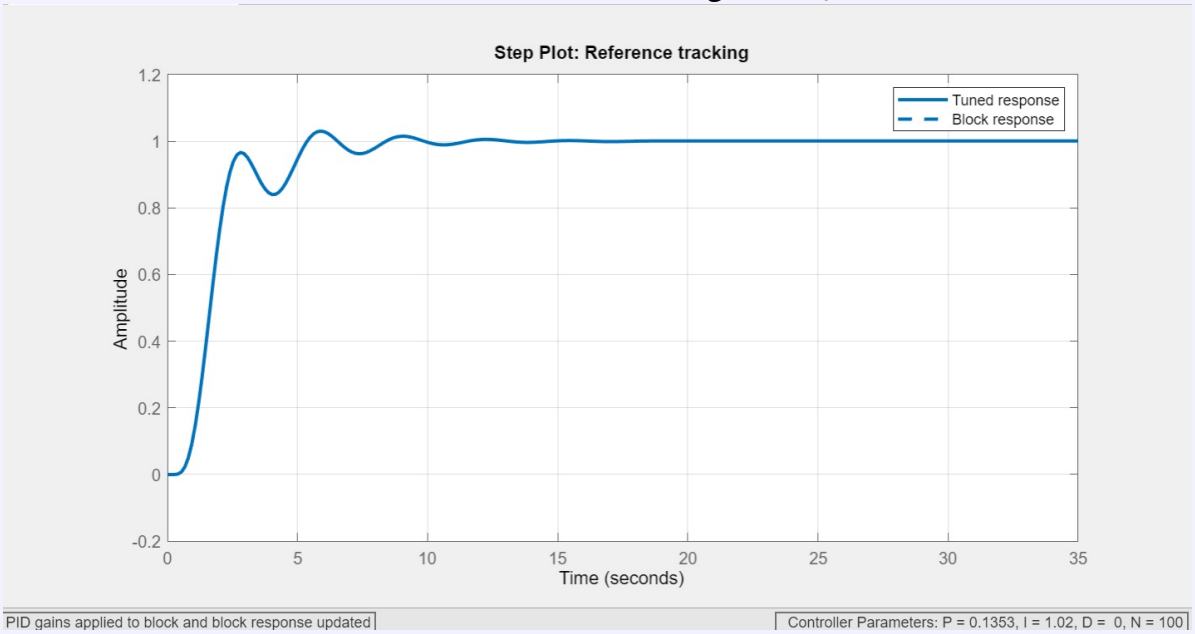
در این قسمت نیز با استفاده از کنترلر داخلی و قرار دادن آن در مدل سیمولینک با نام real_with_controller، با استفاده از گزینه tune بلوک PID و در نهایت fine tuning کردن، به نتایج زیر رسیدیم:



شکل ۱۳: مدل سیمولینک

$$PID = P + I\frac{1}{s} + D\frac{N}{1 + N\frac{1}{s}} \rightarrow \ P = 0.135,\ I = 1.02,\ D = 0,\ N = 100$$

رفتار کنترلر در PID-Tuner به صورت زیر می‌باشد:



شکل ۱۴: پاسخ پله کنترلر

پارامترهای کنترلر نیز به این صورت می‌باشد:

Controller Parameters

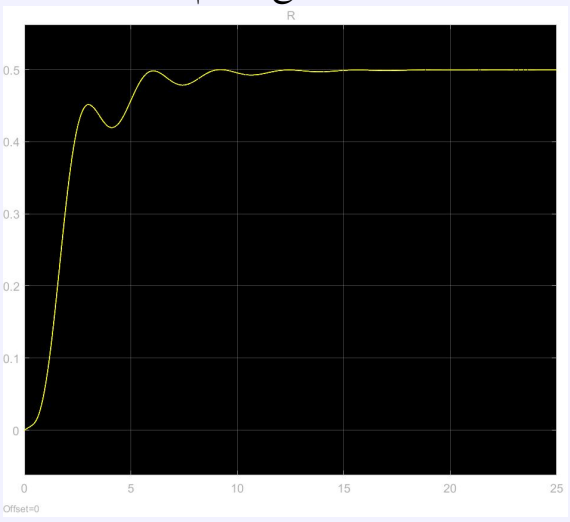
	Tuned	Block
P	0.13534	0.13534
I	1.0196	1.0196
D	0	0
N	100	100

Performance and Robustness

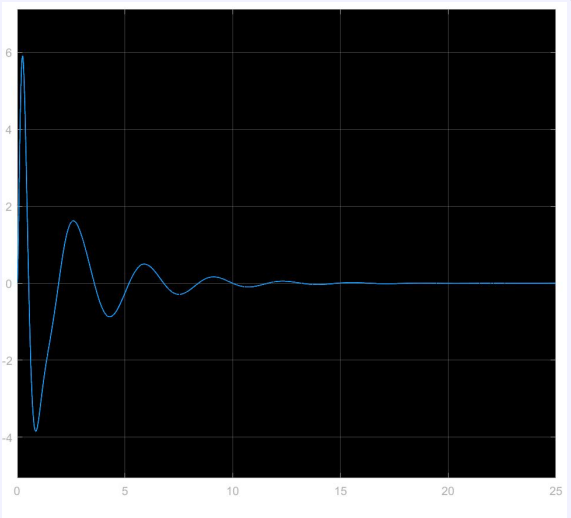
	Tuned	Block
Rise time	1.43 seconds	1.43 seconds
Settling time	7.99 seconds	7.99 seconds
Overshoot	3.03 %	3.03 %
Peak	1.03	1.03
Gain margin	6.06 dB @ 1.92 rad/s	6.06 dB @ 1.92 rad/s
Phase margin	72.9 deg @ 0.547 rad/s	72.9 deg @ 0.547 rad/s
Closed-loop stability	Stable	Unstable

شکل ۱۵: پارامترهای کنترلر

همانطور که مشاهده می‌شود، جعبه‌ابزار به دلیل اینکه حذف صفر و قطب در مدار داخلی را تشخیص نمی‌دهد، به اشتباه آن را ناپایدار گزارش می‌کند ولی سیستم پایدار می‌باشد. همچنین زمان نشست کنترلر زیر ۸ ثانیه بوده و اورشوت زیر ۲۰ درصد می‌زند. حال پاسخ سیستم واقعی را بررسی می‌کنیم:



شکل ۱۶: پاسخ پله



شکل ۱۷: سیگنال کنترلی

همانطور که مشاهده می‌شود اورشوت زیر ۲۰ درصد می‌باشد و سیگنال کنترلی به حد اشباع نرسیده و نیازی به سیستم آنتی ویندآپ نیست. ولی از لحاظ زمان نشست، کنترلر با زمان نشست حدود ۸.۰۳ ثانیه، تا حدودی خواسته ما را ارضاء می‌کند. ولی اگر معیار را به ۵ درصد افزایش دهیم، به راحتی زمان نشست زیر ۸ ثانیه می‌باشد.

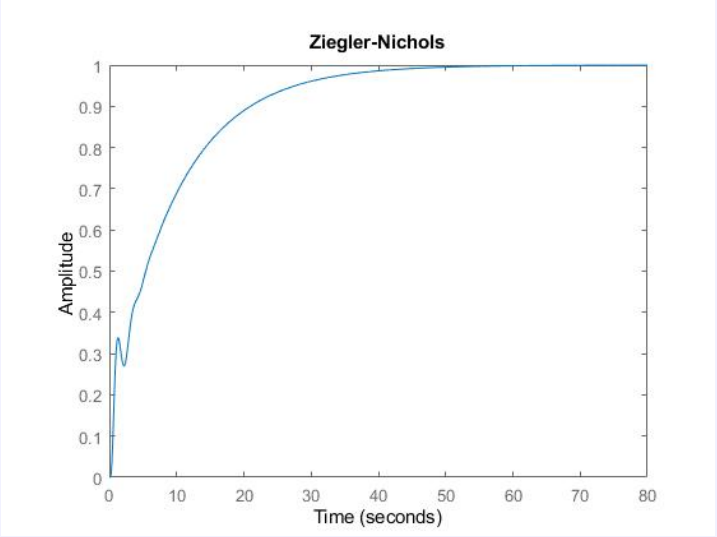
۴. با بکارگیری روش‌های تدریس شده (زیگلر نیکولز، آستروم هاگلند و ...) کنترلر PID مناسب را طراحی کنید.

پاسخ

فایل‌های مربوط به این بخش در فولدر q۴ قرار دارند.

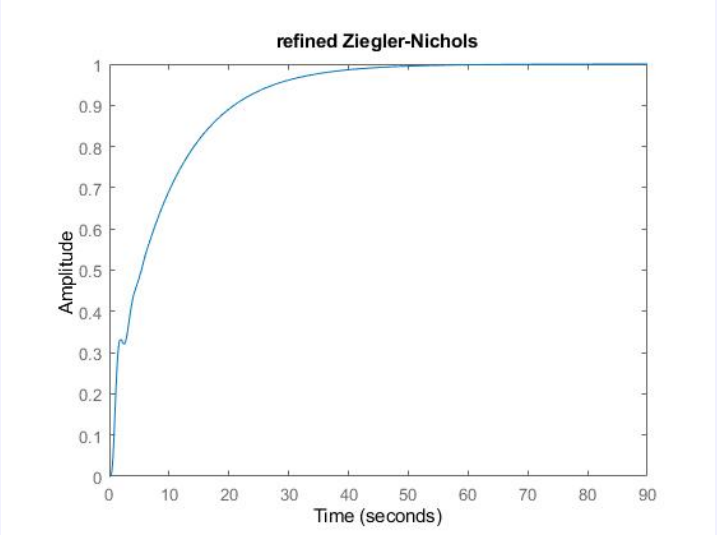
توابع تبدیل و نمودارهای این بخش با استفاده از لایو اسکریپت main قابل بازتولید می‌باشند. ابتدا نمودارها را آورده و سپس آنها را تحلیل می‌کنیم:

ZN



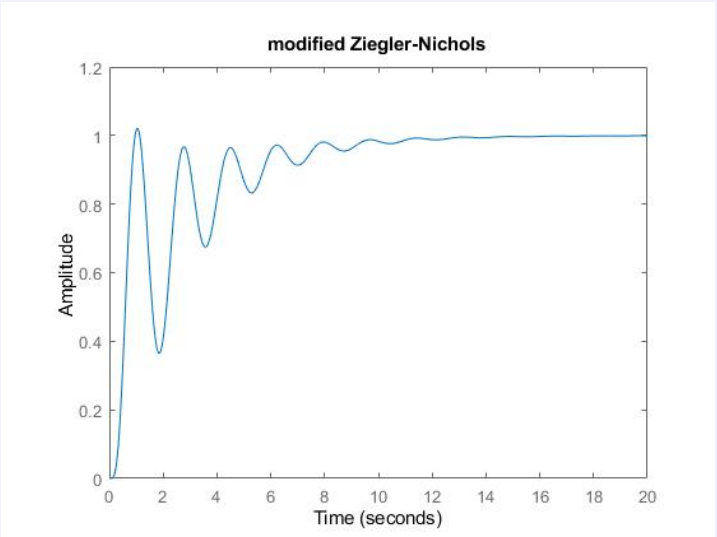
شکل ۱۸: زیگلر نیکولز

Refined ZN



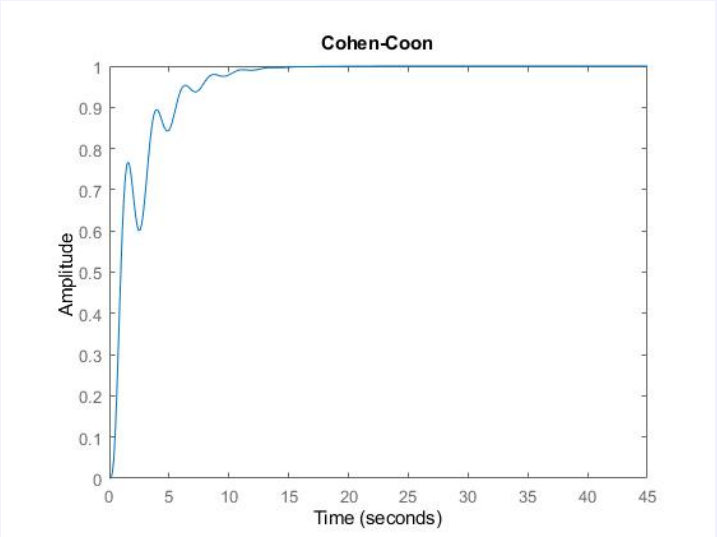
شکل ۱۹: زیگلر نیکولز تقویت شده

Modified ZN



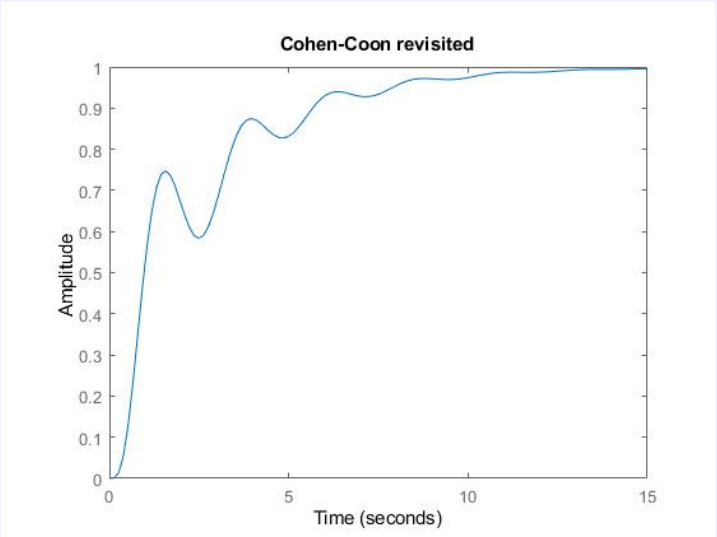
شکل ۲۰: زیگلر نیکولز تصحیح شده

Cohen-Coon



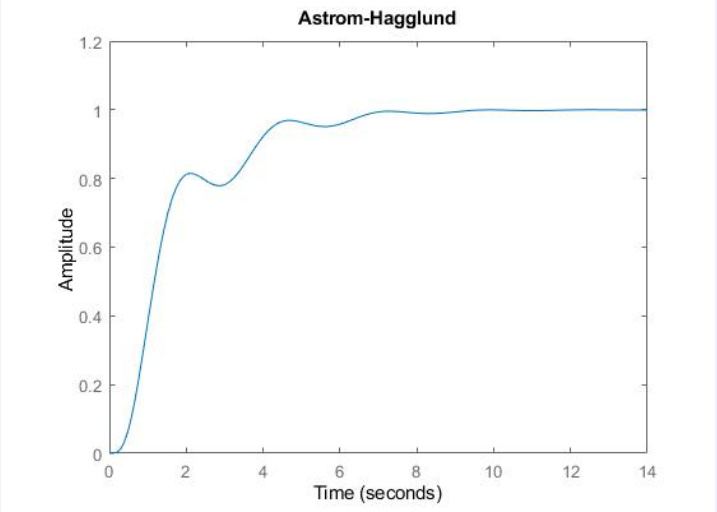
شکل ۲۱: کوهن کون

Revisited Cohen-Coon



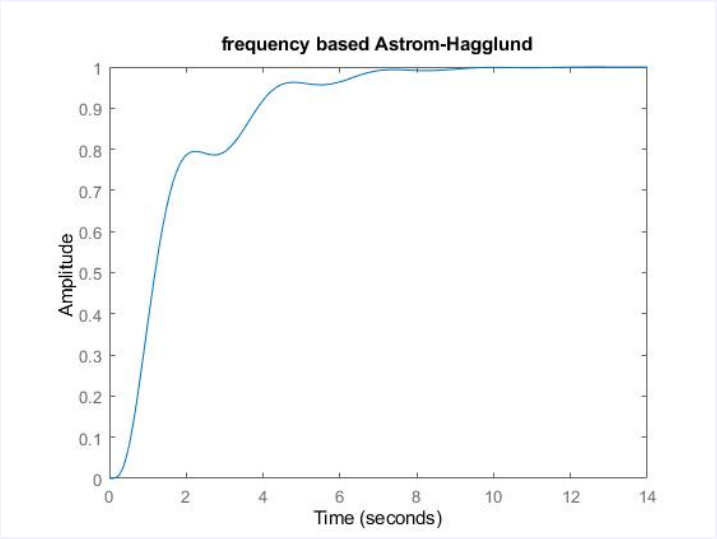
شکل ۲۲: کوهن کون تصحیح شده

Astrom-Hagglund



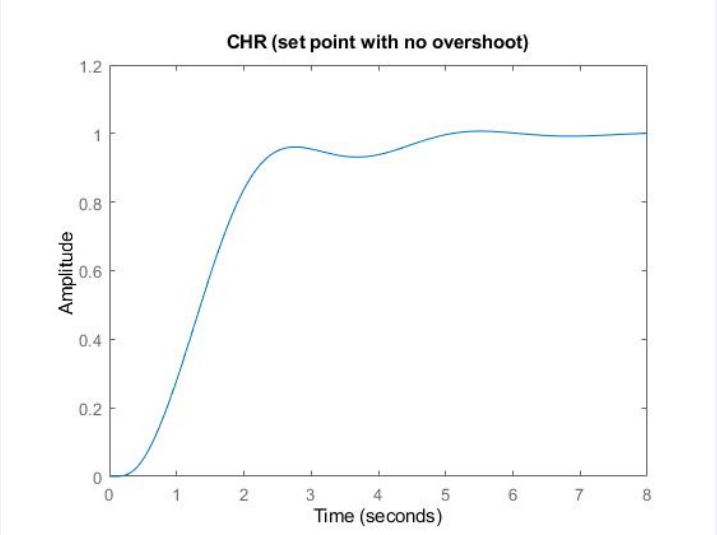
شکل ۲۳: آستروم هاگلند

- Astrom-Hagglund Freq.



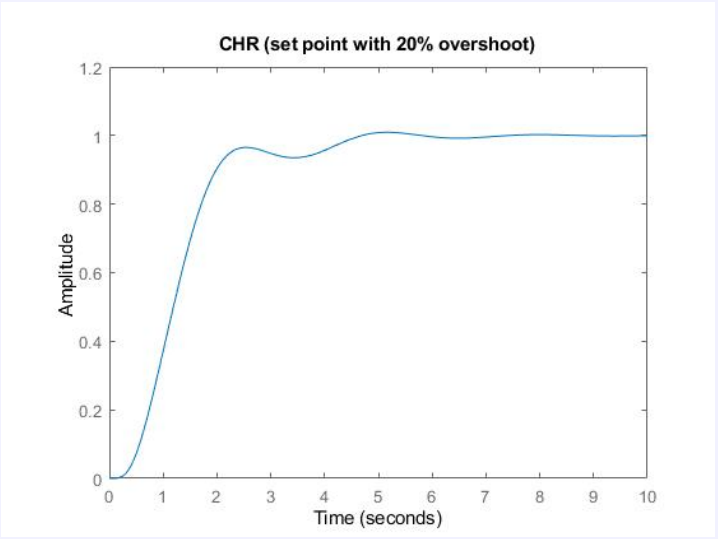
شکل ۲۴: آستروم هاگلند فرکانسی

- CHR Set-Point ۰% Overshoot



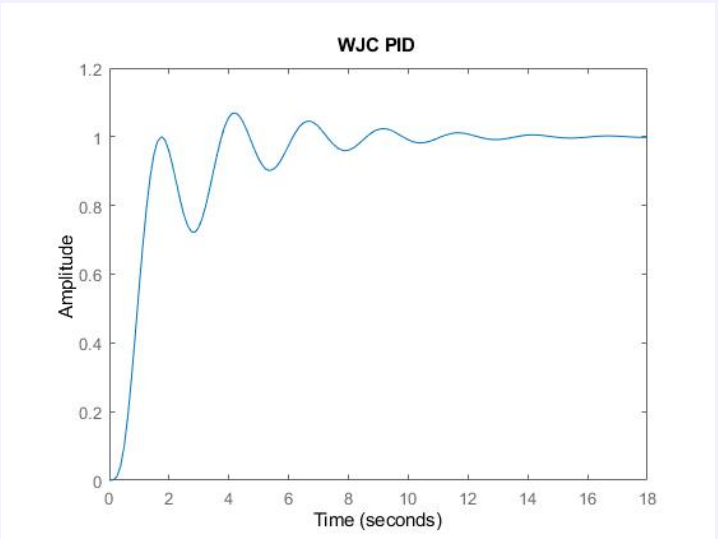
شکل ۲۵: CHR Set-Point ۰% Overshoot

CHR Set-Point ۲۰% Overshoot



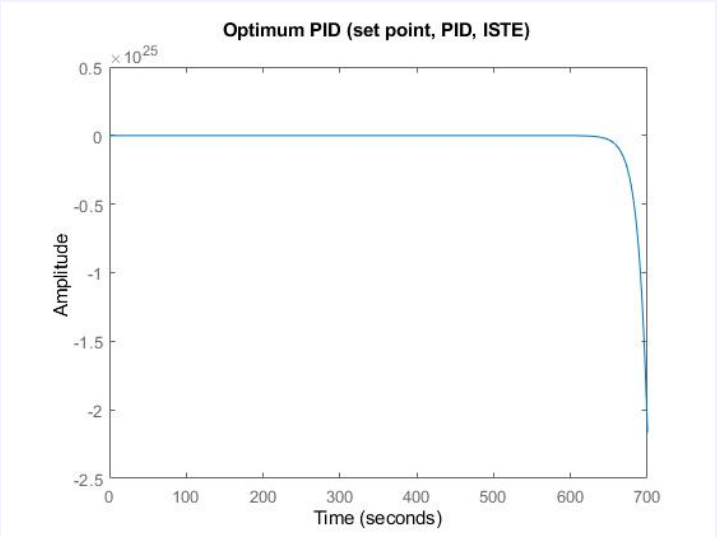
شکل ۲۶: CHR Set-Point ۲۰% Overshoot

WJC PID



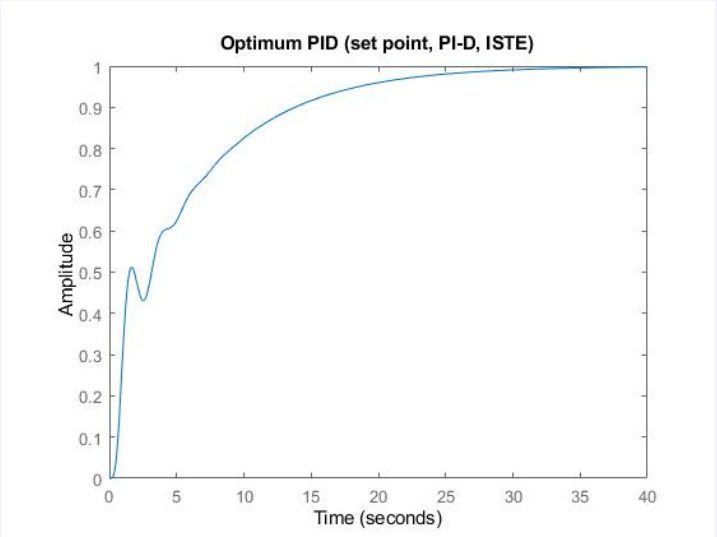
شکل ۲۷: WJC PID

OptimumPID Set-Point PID ISTE



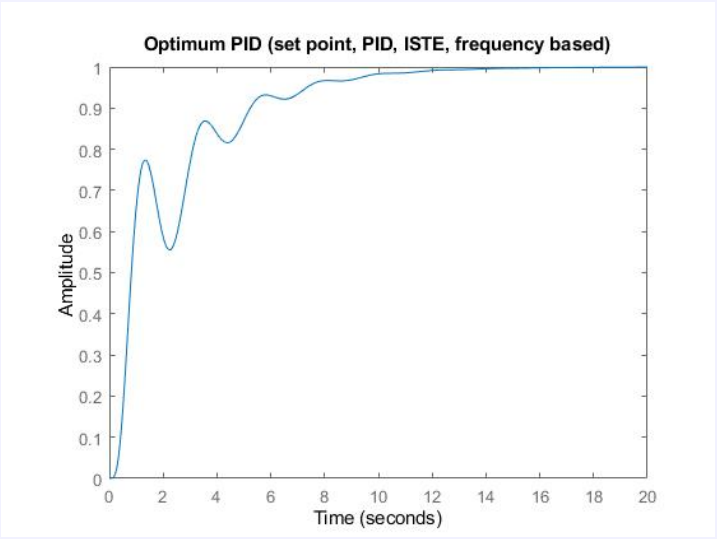
شکل ۲۸: OptimumPID Set-Point PID ISTE

OptimumPID Set-Point PI-D ISTE



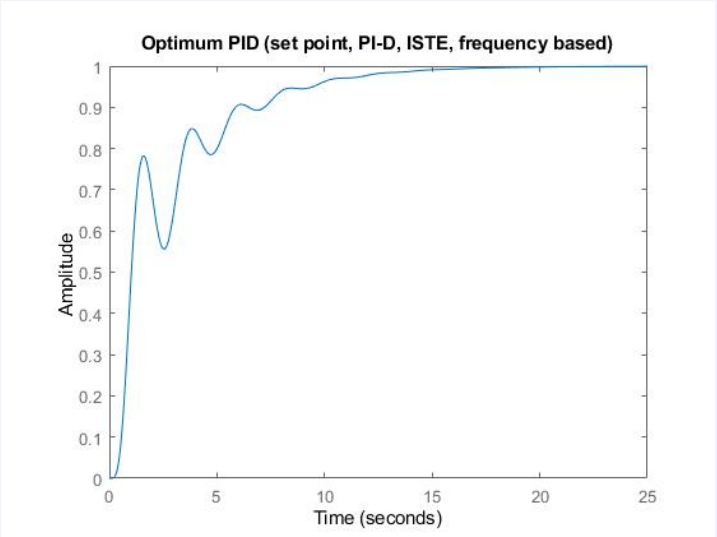
شکل ۲۹: OptimumPID Set-Point PI-D ISTE

- OptimumPID Set-Point PID ISTE Freq.



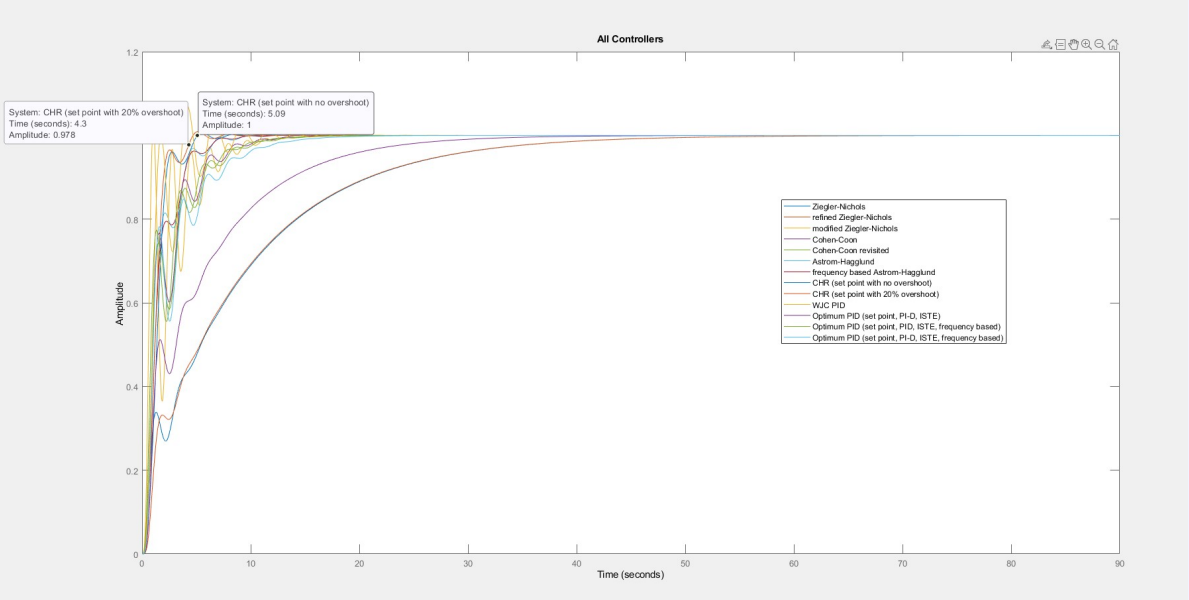
شکل ۳۰: OptimumPID Set-Point PID ISTE Freq.

- OptimumPID Set-Point PI-D ISTE Freq.



شکل ۳۱: OptimumPID Set-Point PI-D ISTE Freq.

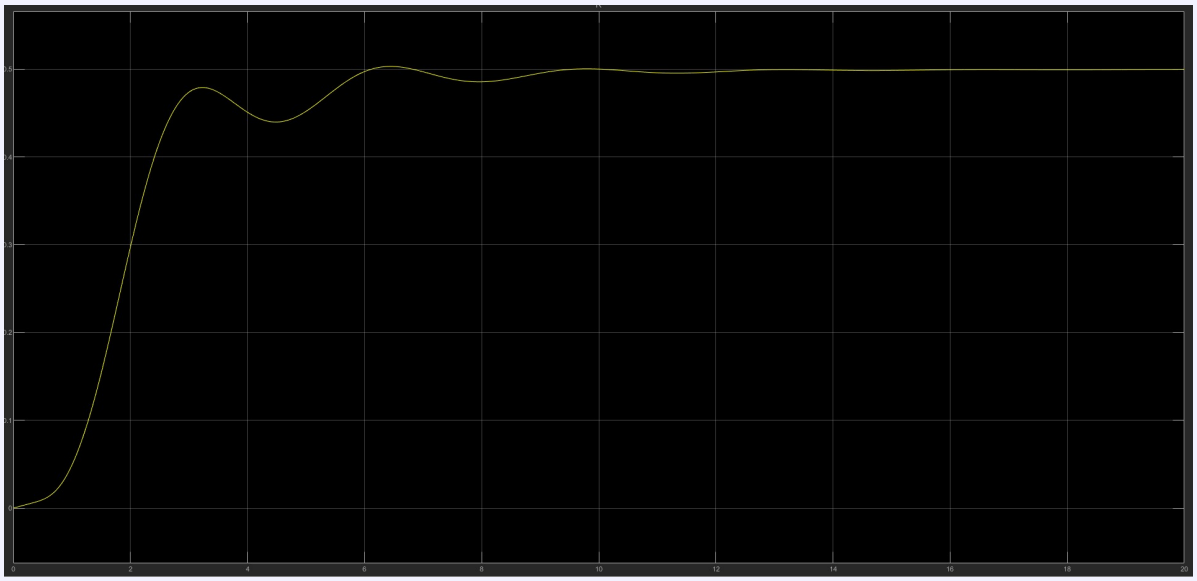
همگی



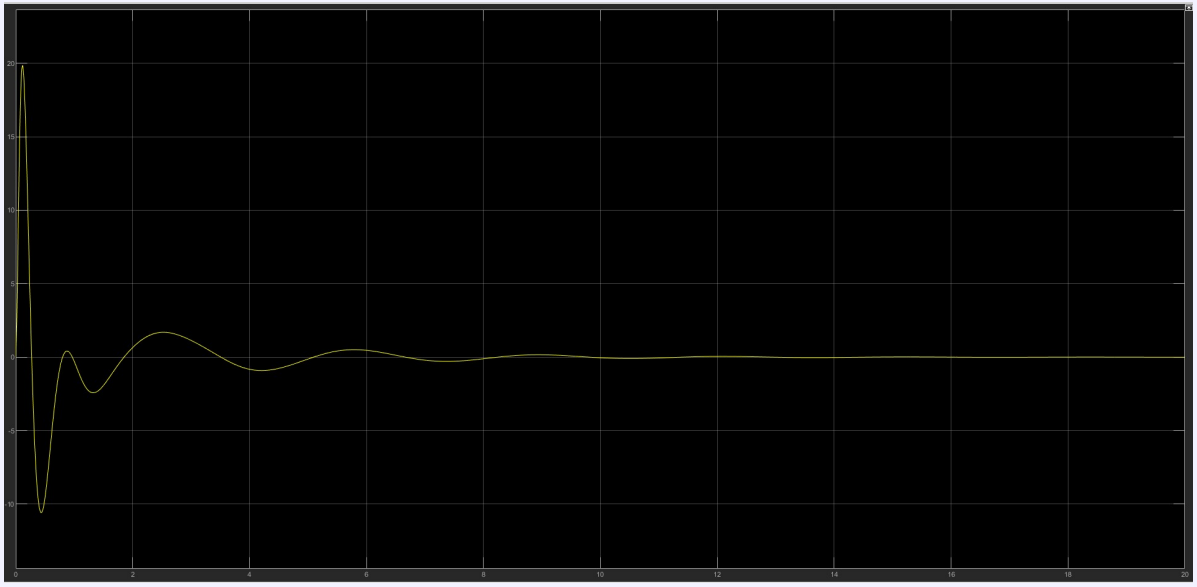
شکل ۳۲: تمامی کنترلرها کنار هم

تحلیل

با توجه به نمودار Ziegler-Nicholes می‌توان مشاهده کرد که اوورشوت نداریم اما زمان نشست در حدود ۴۰ ثانیه است که فراتر از مقدار قابل قبول است. همین رفتار در Refined ZN نیز مشاهده می‌شود. زمان نشست با معیار Modified ZN در مرز ۸ ثانیه قرار دارد اما به علت آندرشوت زیاد از آن صرف نظر می‌شود.
روش Cohen-Coon زمان نشست نزدیک به ۱۲ ثانیه دارد که فراتر از مقدار خواسته شده است. این زمان در شکل Revisited Cohen-Coon نیز وجود دارد. روش Astrom-Haggland ساده و روش بر اساس فرکانس آن نیز زمان نشست ۸ ثانیه دارد و زمان آندرشوت آن کم است، پس یکی از گزینه های مورد نظر است. روش CHR چه بدون اوورشوت و چه با اوورشوت ۲۰ درصد رفتار قابل قبولی دارد و زمان نشست آن کمتر از ۶ ثانیه است.
روش WJC PID زمان نشست ۱۲ ثانیه دارد و همچنین نوسانات آن زیاد است. Optimum PID ناپایدار است و PI-D این روش زمان نشست بالایی دارد هر چند که روش های فرکانسی آن زمان نشست ۱۲ ثانیه دارند اما قابل قبول نیستند.
در نهایت با مقایسه تمام روش ها می‌توان مشاهده کرد که CHR ها از Astrom Haggland بهتر عمل می‌کنند و در نتیجه کنترلر PID انتخاب شده می‌باشند. مدل CHR بدون اوورشوت را با Tuner PID تیون کرده و در نهایت رفتار واقعی مدل سیستم با کنترلر ذکر شده در شکل response system real قابل مشاهده است. (مدل ذکر شده در system۲.slx قرار دارد.)



شکل ۳۳: پاسخ سیستم



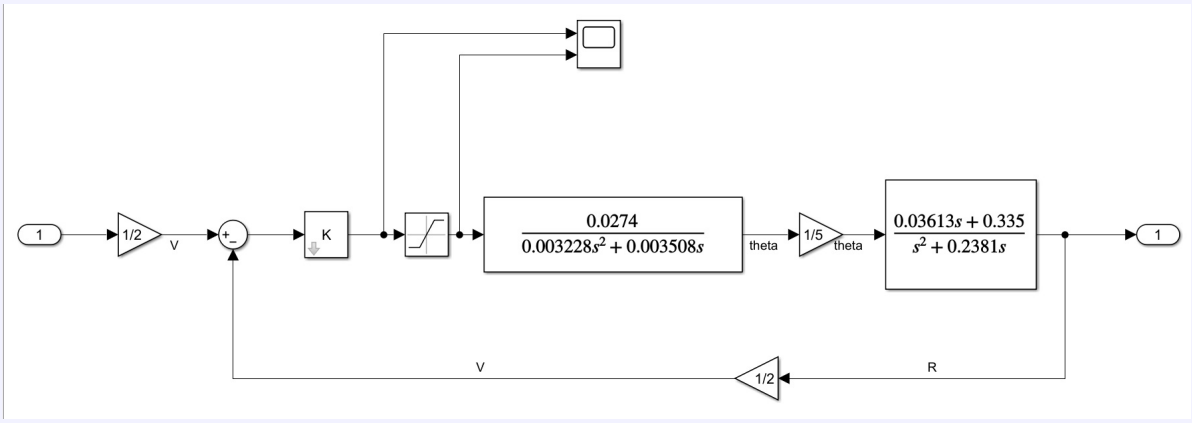
شکل ۳۴: سیگنال کنترلی

۵. با استفاده از ابزار optim pid کنترلری از خانوادهی PID طراحی کنید بگونهای که شرایط فوق حاصل گردد.

پاسخ

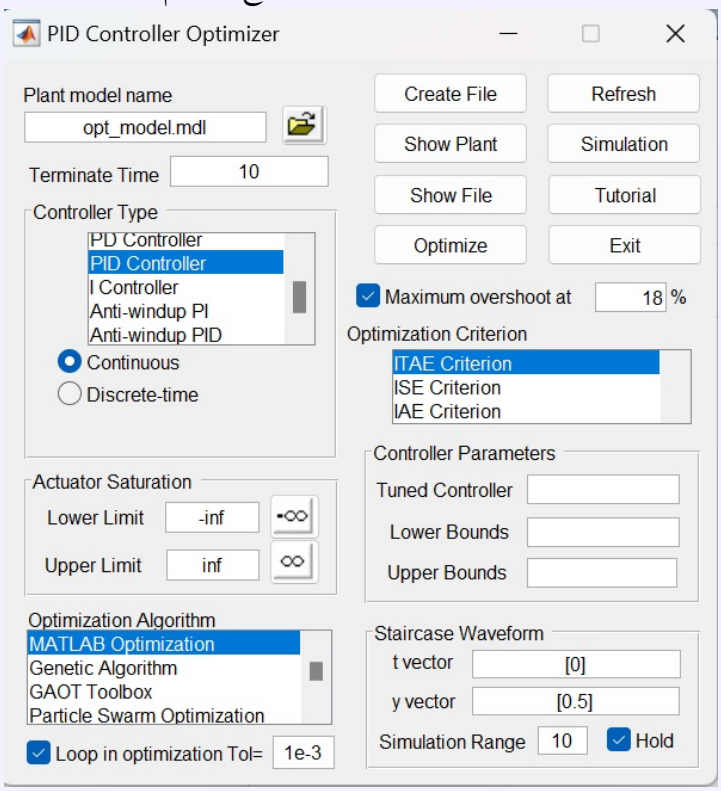
فایل‌های مربوط به این بخش در فولدر q۵ قرار دارند.

در این بخش برای استفاده از ابزار OptimPID، نیاز داریم مدل زیر را بسازیم که با نام opt_model در فولدر قرار دارد:



شکل ۳۵: مدل برای بهینه‌سازی

حال با قرار دادن پارامترهای بهینه‌سازی به صورت زیر، روند را شروع می‌کنیم:

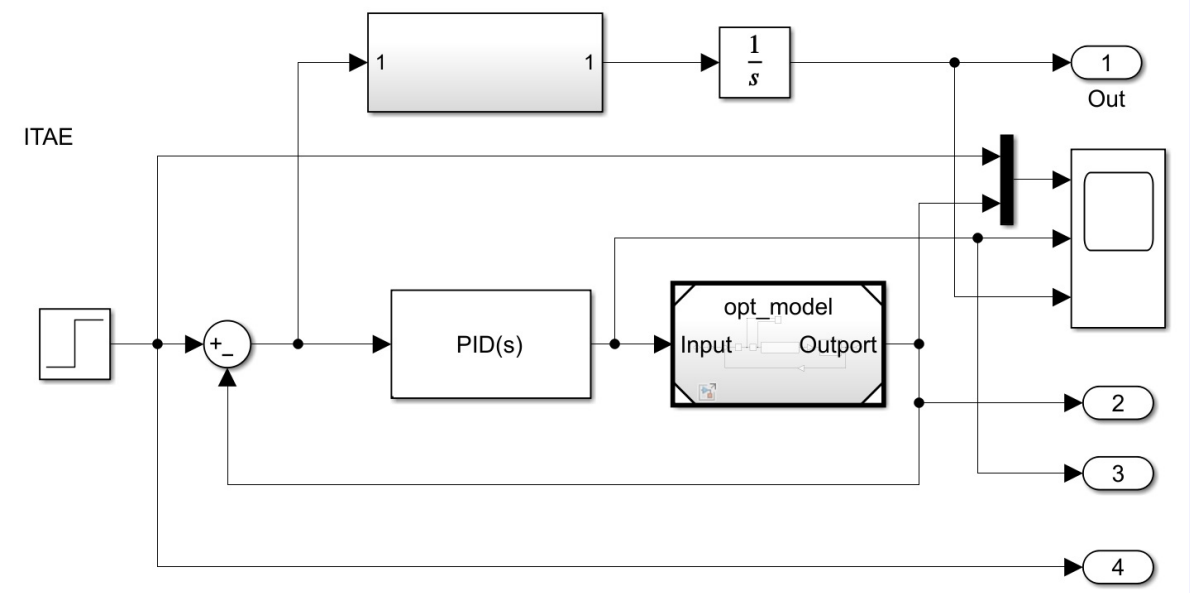


شکل ۳۶: پارامترهای بهینه‌سازی

مقادیر به دست آمده به صورت زیر می‌باشند:

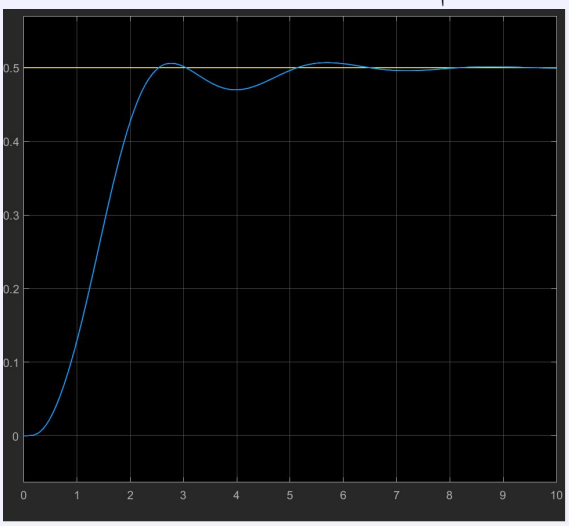
$$PID = P + I\frac{1}{s} + D\frac{N}{1 + N\frac{1}{s}} \rightarrow P = 0.159,\ I = 1.380,\ D = 0.0889\ N = 100$$

حال رفتار سیستم اصلی را به ازای این کنترلر بررسی می‌کنیم که سیمولینک مربوطه با نام without_anti_windup ذخیره شده است:

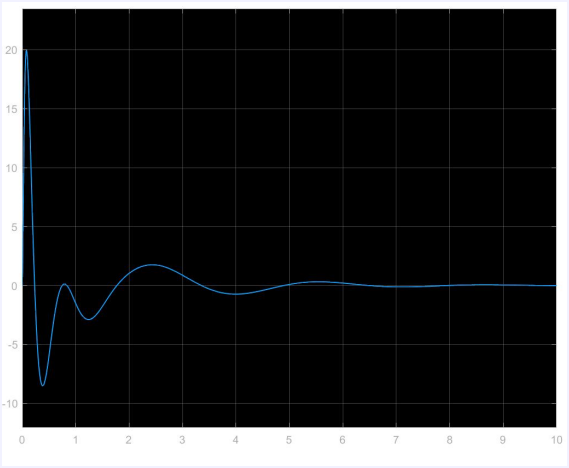


شکل ۳۷: سیمولینک

رفتار سیستم به صورت زیر می‌باشد:



شکل ۳۸: پاسخ پله



شکل ۳۹: سیگنال کنترلی

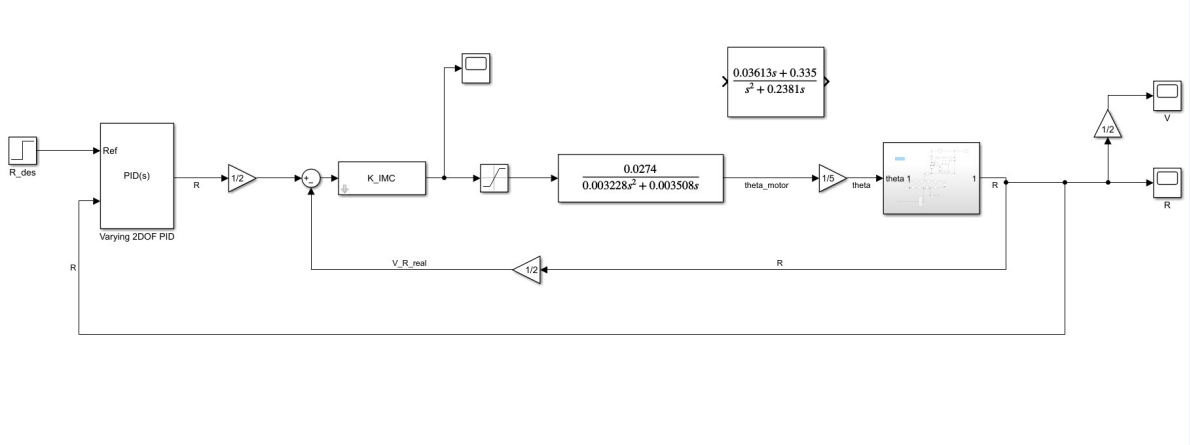
همانطور که مشاهده می‌شود زمان نشست کمتر از ۸ ثانیه، و اورشوت نیز زیر ۲۰ درصد می‌باشد. همچنین سیگنال کنترلی نیز دچار اشباع نمی‌شود. در نتیجه این کنترلر همه‌ی خواسته‌های ما را ارضاء می‌کند.

۶. یک کنترلر PID دو درجه آزادی برای رسیدن به خواسته‌های مسئله طراحی کنید و این کنترلر را با کنترلرهای طراحی شده در قسمت‌های قبلی مقایسه نمایید.

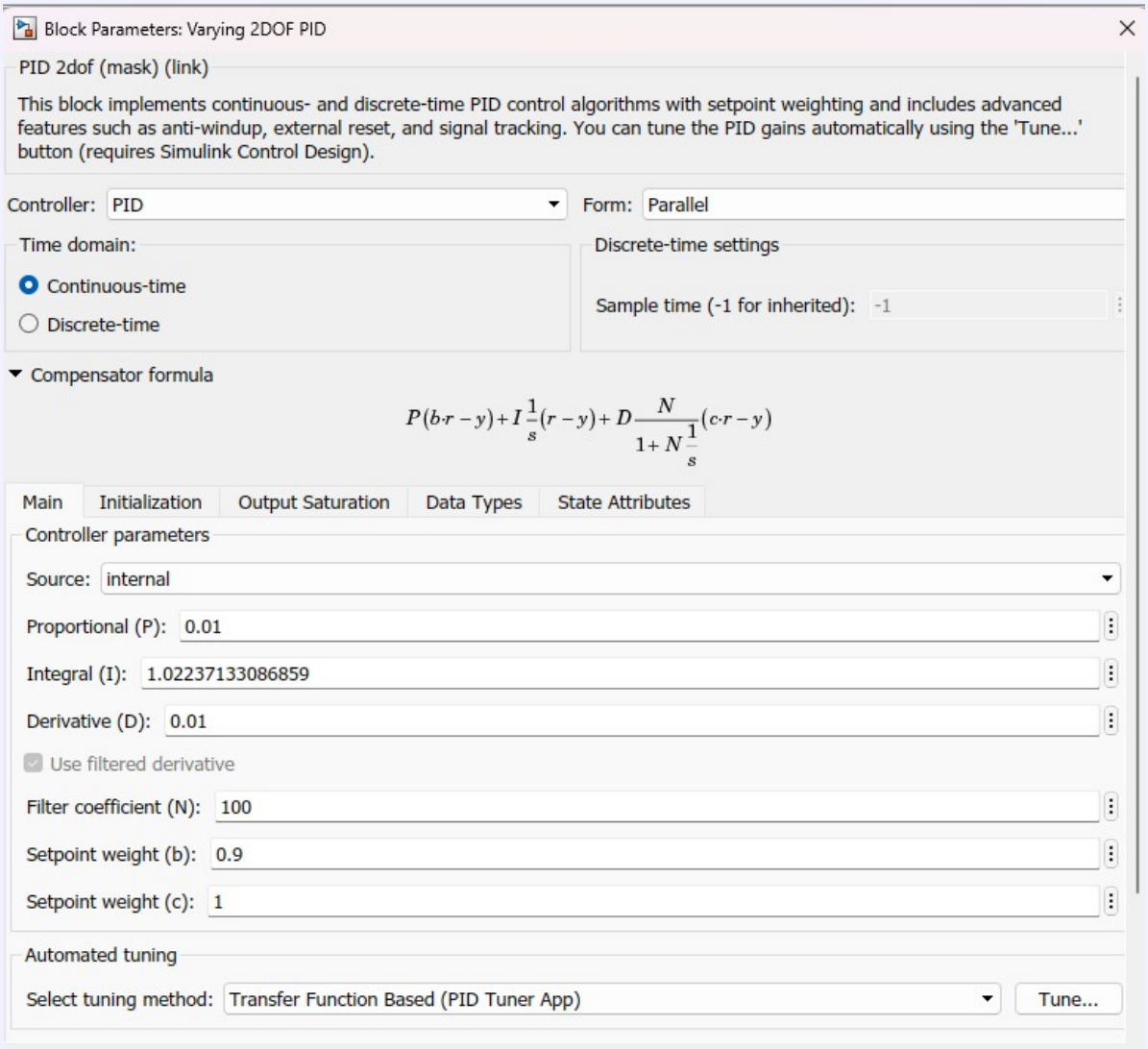
پاسخ

فایل‌های مربوط به این بخش در فولدر q۶ قرار دارند.

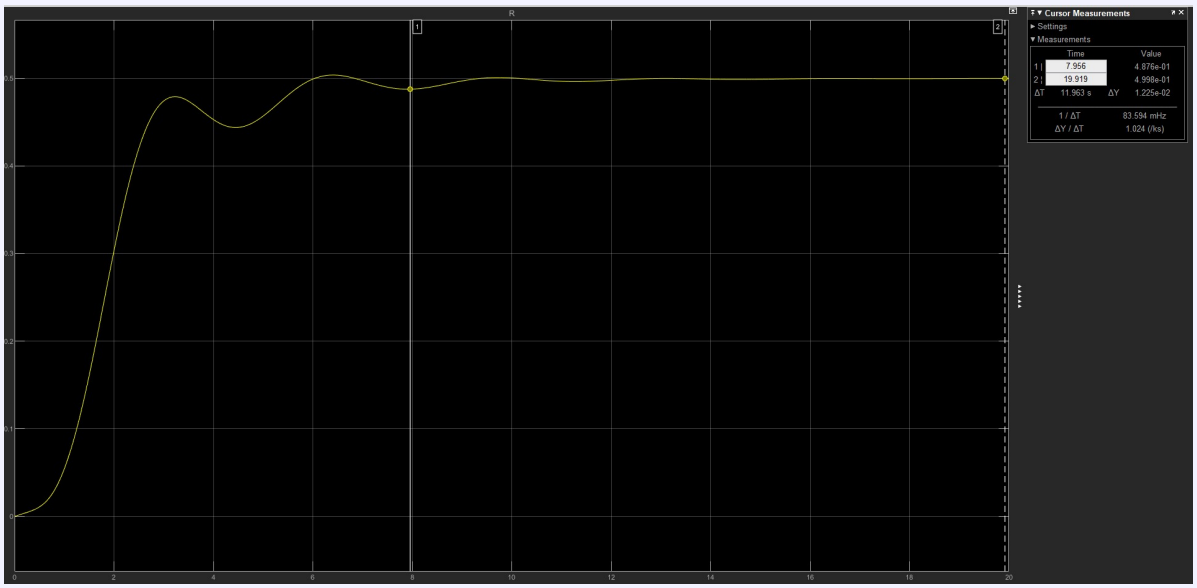
در مدل simulink۳.slx از از PID دو درجه آزادی استفاده می‌کنیم.پس از تیون کردن ضرایب، به مقدار نشست ۸ ثانیه و اوورشوت ۲ درصد می‌رسیم که ضرایب آن ها قابل مشاهده است.



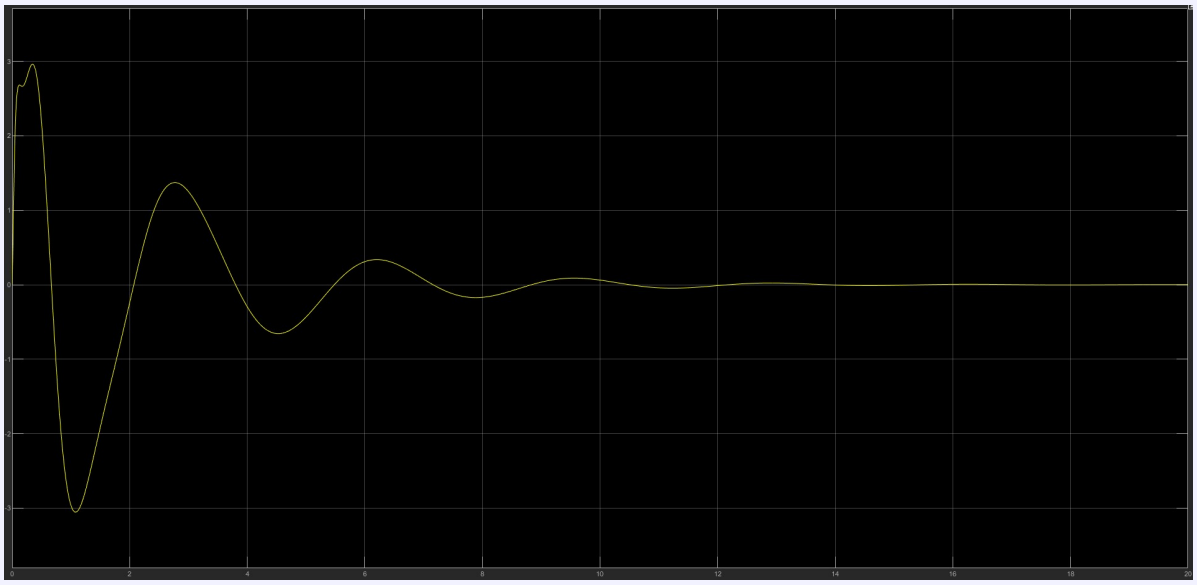
شکل ۴۰: مدل سیمولینک



شکل ۴۱: پارامترهای کنترلر



شکل ۴۲: رفتار سیستم اصلی



شکل ۴۳: سیگنال کنترلی

۷. کدامیک از کنترلرهای طراحی شده در قسمت‌های قبل، نسبت به اغتشاشی با فرکانس ۳۰ هرتز مقاوم است؟ اگر هیچکدام از کنترلرهای قبلی این شرایط را ندارند، کنترلری (به روش دلخواه) طراحی کنید که علاوه بر خواسته‌های گفته شده، دامنه‌ی نوسانات نهایی را به کمتر از ۳ درصد دامنه‌ی اغتشاش برساند.

پاسخ

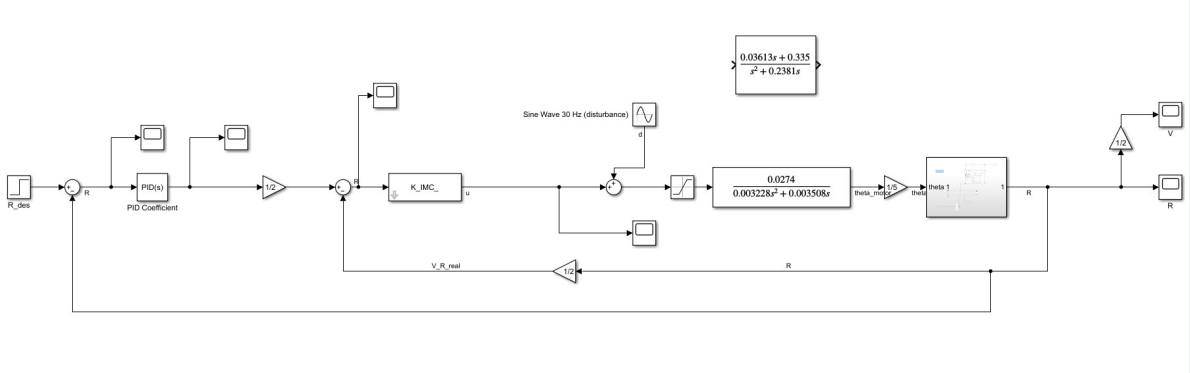
فایل‌های مربوط به این بخش در فولدر q۷ قرار دارند.

در این سوال در صورت استفاده از کنترلر داخلی K به خاطر وجود اغتشاش، آفستی حدود ۱۵ درصد خواهیم داشت برای همین با روش یولا کوچرا و استفاده از Q=۰ به کنترلری می¬رسیم که نسبت به اغتشاش مقاوم است. کنترلر داخلی

جدید به صورت زیر می‌باشد: (کنترلر زیر را می‌توان از بخش مربوط به سوال ۸ لایو اسکریپت main بازتولید کرد.)

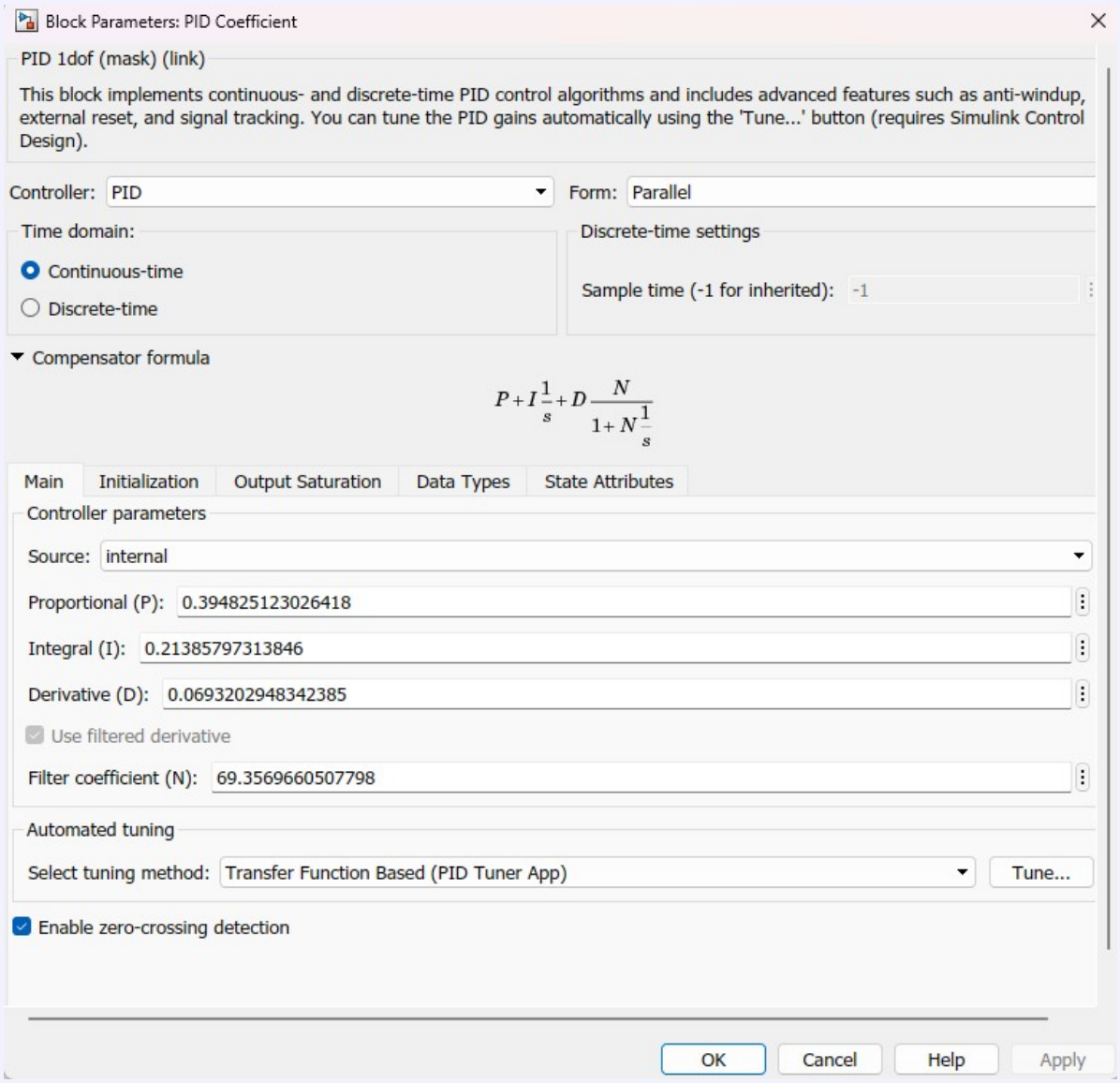
$$K_{IMC} = \frac{35.27s^3 + 57.66s^2 + 24.24s + 3.517}{s^3 + 5.676s^2 + 13.22s + 14.93}$$

حال با کنترلر جدید، مدل سیمولینک زیر را تشکیل می‌دهیم:(فایل سیمولینک مربوطه به اسم system۴ می‌باشد.)

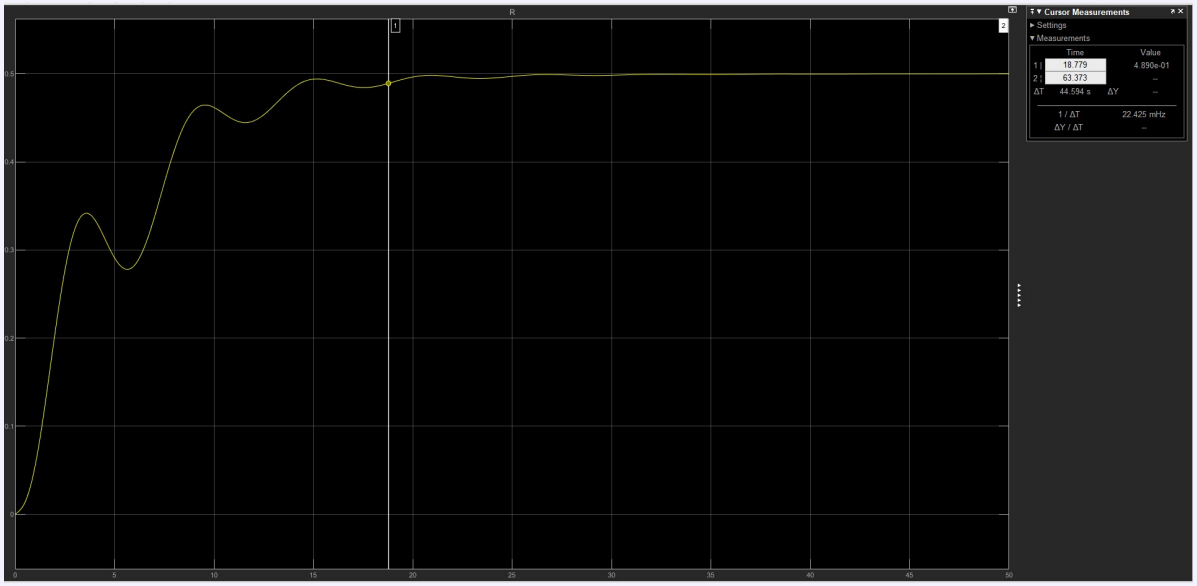


شکل ۴۴: مدل سیمولینک

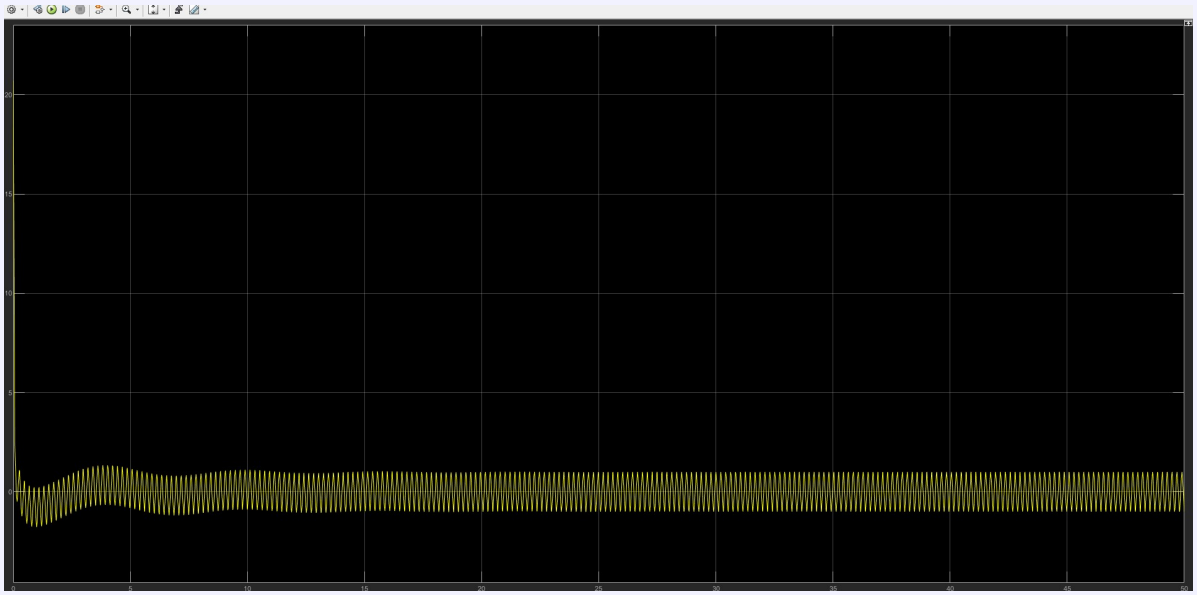
سپس PID مناسب را تیون کرده و در نهایت به پاسخ شکل برای سیستم واقعی می‌رسیم. این سیستم اوورشوت ندارد اما زمان نشست ۱۹ ثانیه ای آن طولانی است.



شکل ۴۵: پارامترهای کنترلر



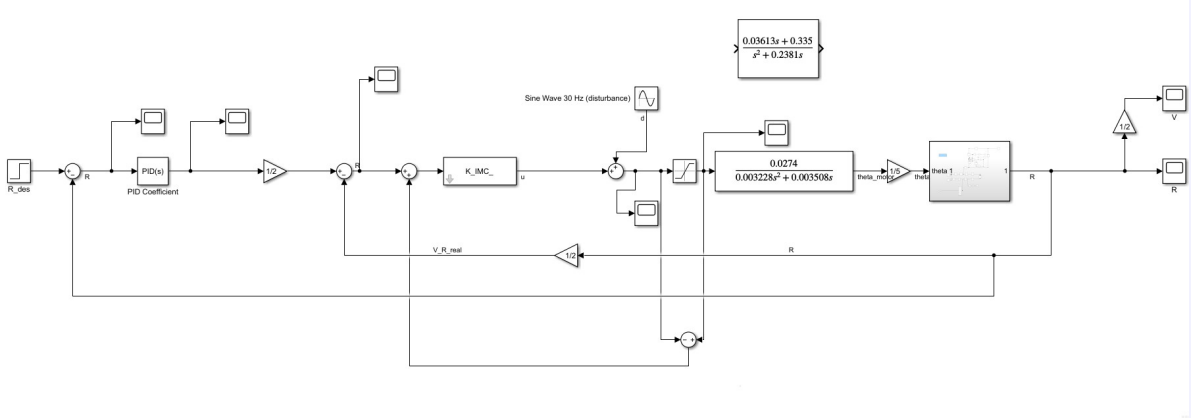
شکل ۴۶: پاسخ پله به همراه اغتشاش



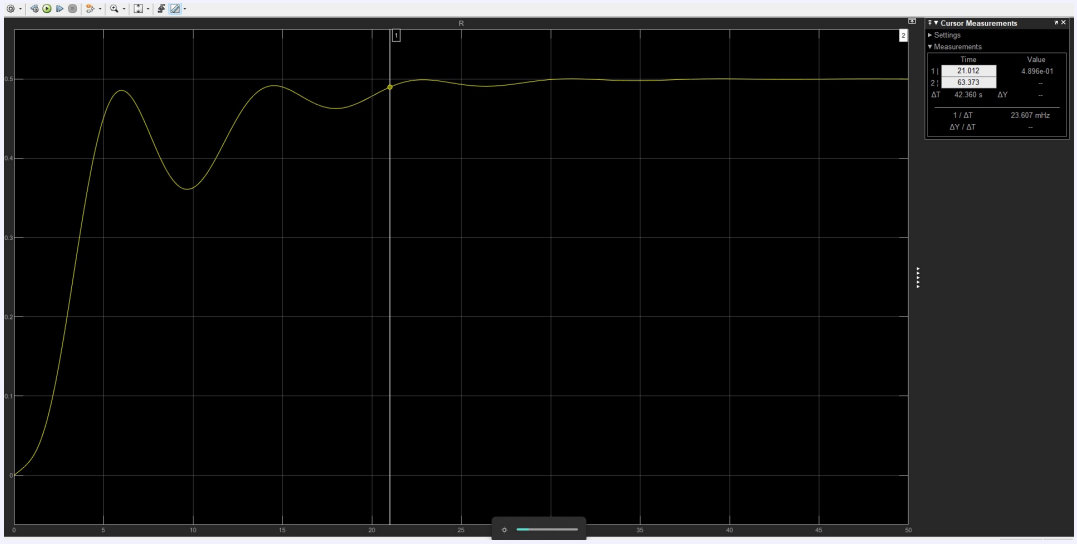
شکل ۴۷: سیگنال کنترلی به همراه اغتشاش

اگر برای سیستم جدید از روش های دیگری برای افزایش سرعت از جمله Lead ، PD و IMC استفاده کنیم سیستم ناپایدار شده و یا اوورشوت بالایی می‌دهد.

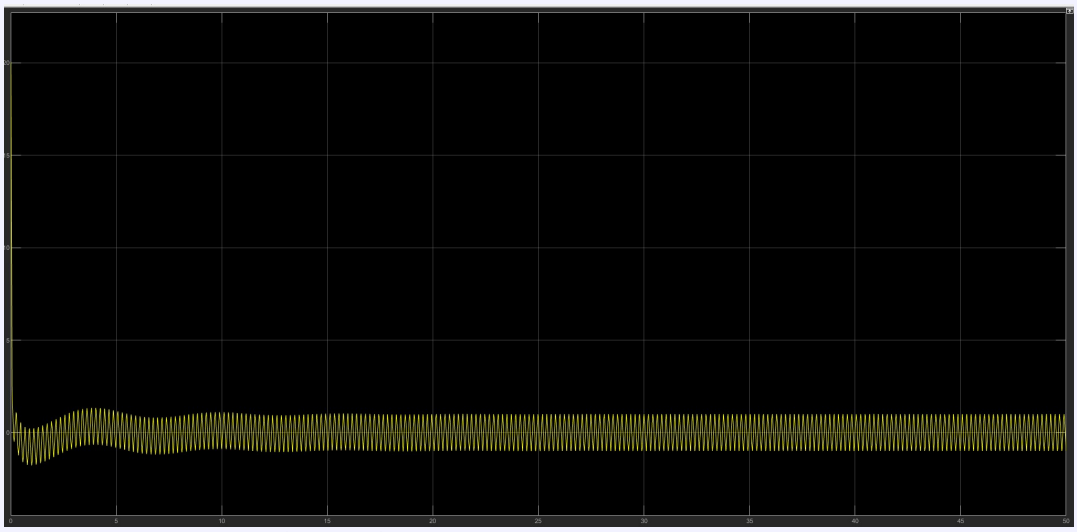
حال یک بار با سیستم آنتی ویندآپ نیز امتحان می‌کنیم. فایل سیمولینک مربوطه به نام system۴_antiwindup در فولدر قرار دارد.



شکل ۴۸: مدل سیمولینک همراه با سیستم آنتی ویندآپ



شکل ۴۹: پاسخ پله همراه با سیستم آنتی ویندآپ



شکل ۵۰: سیگنال کنترلی همراه با سیستم آنتی ویندآپ

همانطوری که مشخص است، سیستم آنتی ویندآپ تأثیر چشمگیری در رفتار سیستم نداشته و خواسته‌های ما را برآورده نمی‌کند. در نتیجه باید راه‌های دیگری را امتحان کرد، مانند طراحی کنترلر داخلی به گونه‌ای که تابع تبدیل مدار بسته حلقه داخلی تا حد خوبی خواسته‌های ما را ارضاء کند.

البته توجه داشته باشید که اغتشاش ورودی دارای دامنه ۱ می‌باشد که دو برابر ورودی می‌باشد و با این حال سیستم به مقدار نهایی رسیده و پایدار است. اگر دامنه‌ی اغتشاش را کم کنم، زمان نشست نیز کم شده، و خواسته زمان نشست نیز ارضاء می‌شود.