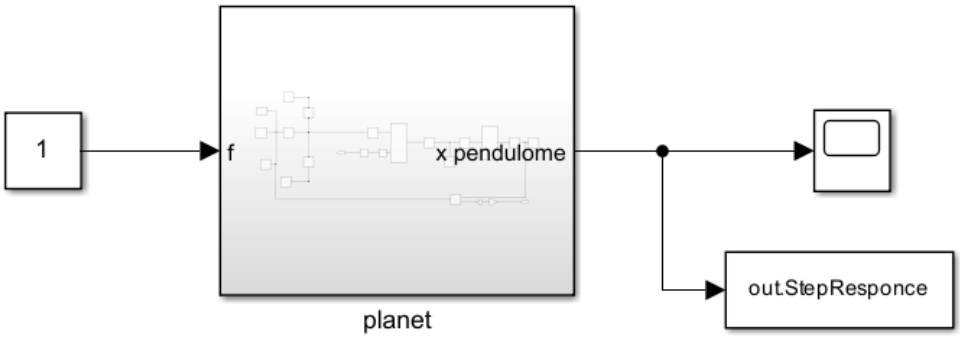
گزارش پروژه کنترل سیستم گاری و پاندول قسمت اول

1 . رسم پاسخ پله سیستم و استخراج مشخصات پاسخ پله و شناسایی یک مدل مرتبه ۲ بر اساس پاسخ پله

ابتدا مدل سیمولینک سیستم را به صورت زیر تغییر میدهیم و به آن ورودی پله اضافه میکنیم و خروجی را در workspace ذخیره میکنیم.



حال مدل را اجرا میکنیم و داده خروجی را با استفاده از تابع stepinfo متلب بررسی میکنیم و مشخصات پاسخ پله به صورت زیر به دست می آیند:

Percent Over Shoot	**38.63 %**
Settling Time	**4.91 s**
Rise Time	**0.46 s**
Peak Time	**1.23 s**
Final Value	**0.05**

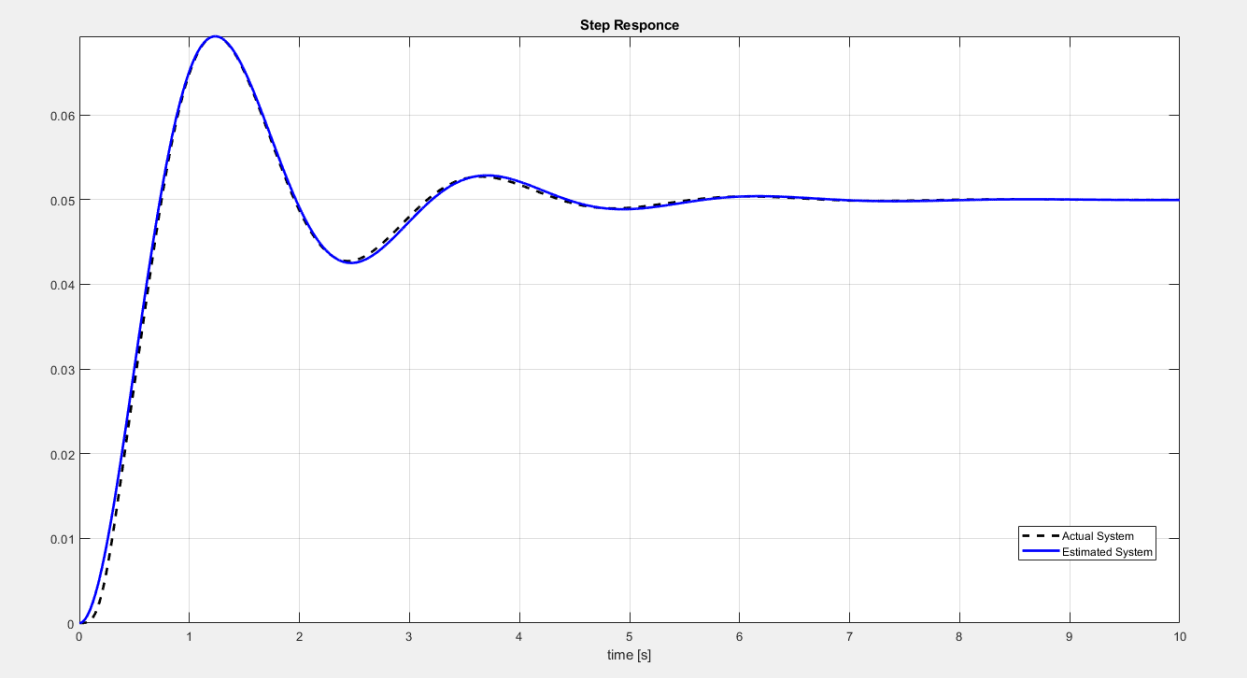
سپس با استفاده از مشخصات بدست آمده میتوانیم یک مدل مرتبه دوم تیپیکال به فرم زیر برای سیستم شناسایی کنیم:

$$G = \frac{\omega_n^2}{S^2 + 2\zeta\omega_n s + \omega_n^2}$$

$$\zeta = \frac{-\log\left(\frac{PO}{100}\right)}{\sqrt{\pi^2 + \left(\log\left(\frac{PO}{100}\right)\right)^2}} \qquad \text{and} \qquad \omega_n = \frac{\pi}{PT\sqrt{1-\zeta^2}}$$

باید دقت داشته باشیم که این سیستم شناسایی شده، یک سیستم با مقدار نهایی برابر ۱ میباشد. برای این منظور باید سیستم شناسایی شده را در مقدار نهایی سیستم اصلی ضرب کنیم تا مقدار مقیاس سیستم ها نیز یکی باشند.

حال پاسخ پله سیستم شناسایی شده را بر روی پاسخ پله اصلی سیستم رسم میکنیم و میبینیم که با دقت بسیار بالایی بهم نزدیک می باشند.



سیستم شناسایی شده را برای قسمت های بعدی در فایل ذخیره میکنیم.

2 . احتساب مقدار حد بهره و حد فاز

برای اینکار یک متد کاملا اتوماتیک پیاده کردیم که حد بهره و حد فاز را به صورت اتوماتیک اندازه گیری میکنند:

۱. ابتدا یک کد برای ران کردن مدل سیمیولینک نوشته و آن را برای مقداری مختلف فرکانس های تحریک به اندازه ۱۰۰ ثانیه ران میکنیم.

۲. ۵۰ ثانیه بر روی پاسخ بدست آمده از سیستم یک بار تبدیل فوریه میزنیم و با استفاده از آن فاز فرکانس تحریک را در خروجی حساب میکنیم.

۳. همچنین با استفاده از ماکسیمم گیری از ۵۰ ثانیه آخر سیگنال خروجی دامنه خروجی را هم حساب میکنیم(از ۵۰ ثانیه آخر استفاده میکنیم تا مطمین باشیم سیستم حالت گذرای خود را رد کرده است.

۴. حال لیست دامنه ها و فازهارا داریم و از روی آن میتوانیم حد فاز و حد بهره را حساب کنیم.

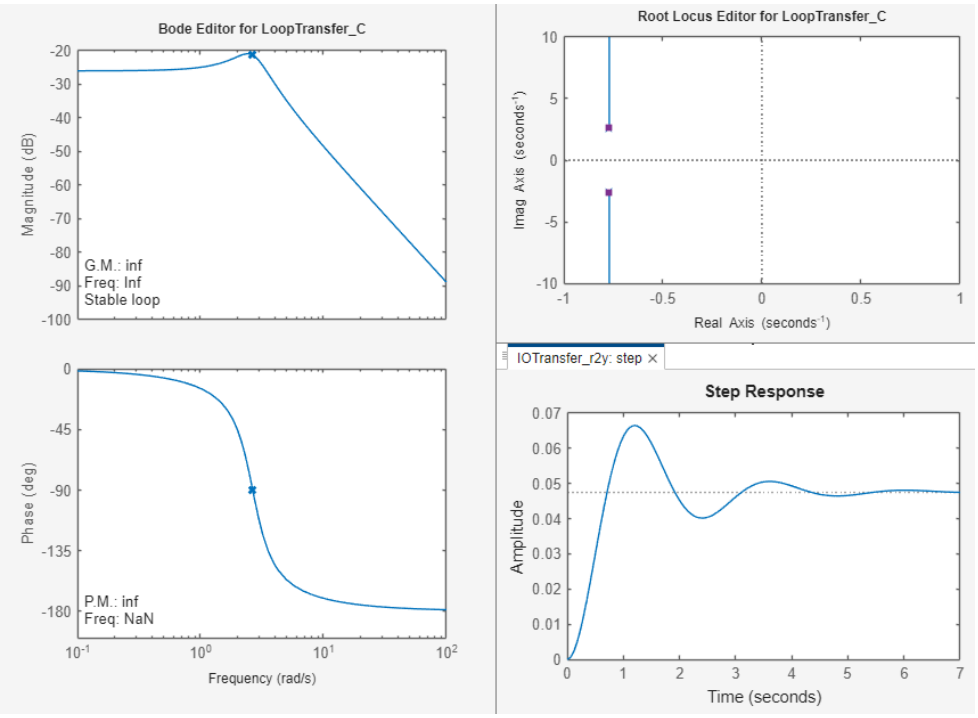
پاسخ نهایی به این صورت میباشد:

Gain Margin (GM): 26.00 dB

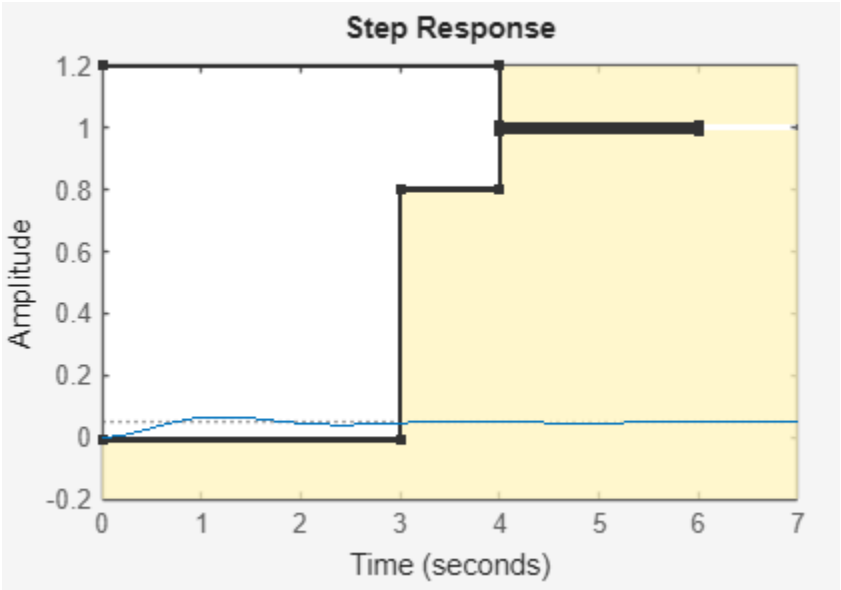
Phase Margin (PM): 58.84 degrees

3. طراحی یک جبران ساز با استفاده از sisotool با مشخصات فراجهش کمتر از ۲۰ درصد و زمان نشست کمتر از ۴ ثانیه بر روی سیستم شناسایی شده

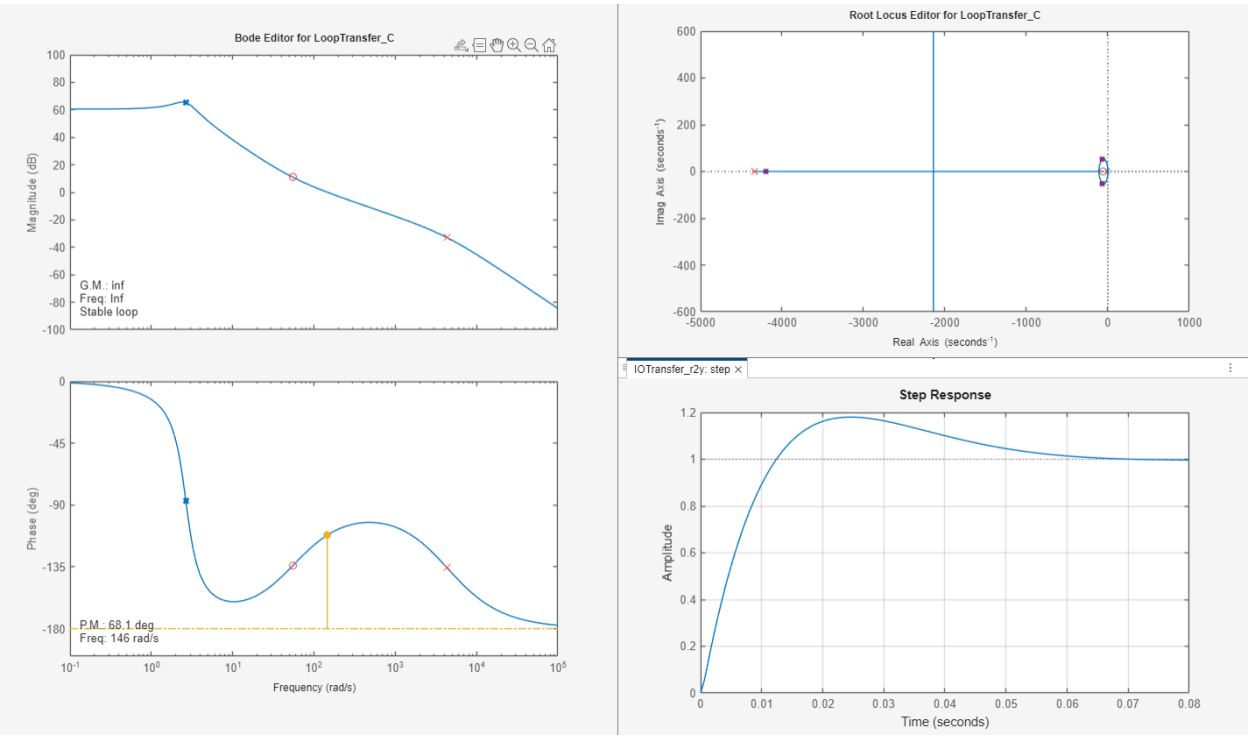
ابتدا سیستم را لود میکنیم و ابزار sisotool را فراخوانی میکنیم.



حال مشخصات خواسته شده را اضافه میکنیم تا محدوده ی مجاز پاسخ پله مشخص شود.

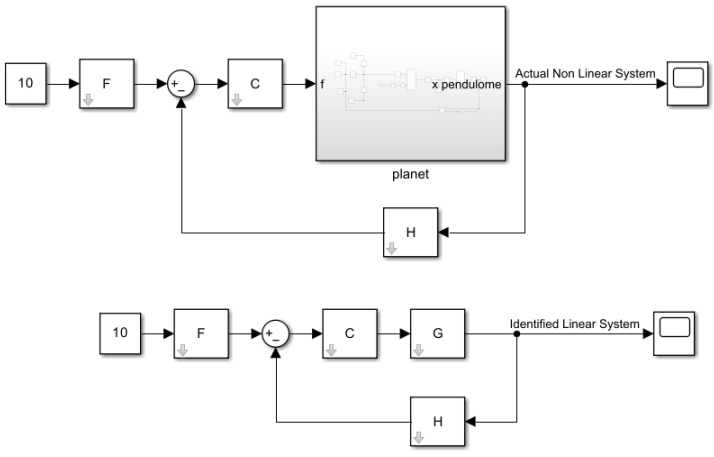


حال با افزودن جبران ساز دلخواه از نوع lead یا lag و دستکاری کردن نمودار بود یا نمودار مکان هندسی ریشه ها میتوانیم کنترل کننده مناسب را طراحی کنیم. و خروجی آن را ذخیره کنیم.



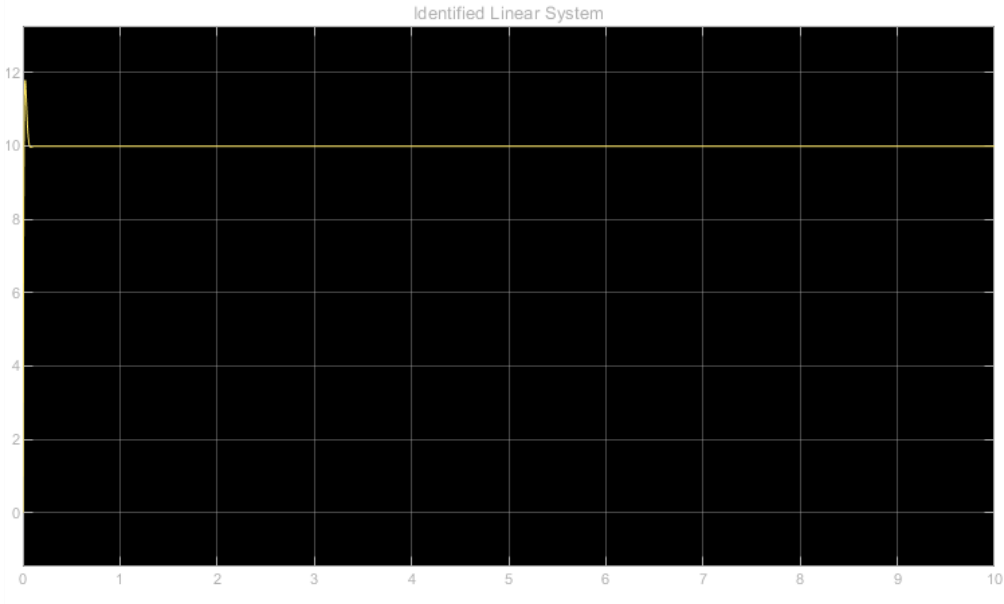
4 . تست کردن جبران ساز طراحی شده در قسمت ۳ بر روی سیستم اصلی و سیستم شناسایی شده

برای این کار کنترل کننده طراحی شده در قسمت ۳ و سیستم شناسایی شده را از قبل لود میکنیم و سیستم سیمیولینک خود را به صورت زیر آماده میکنیم:

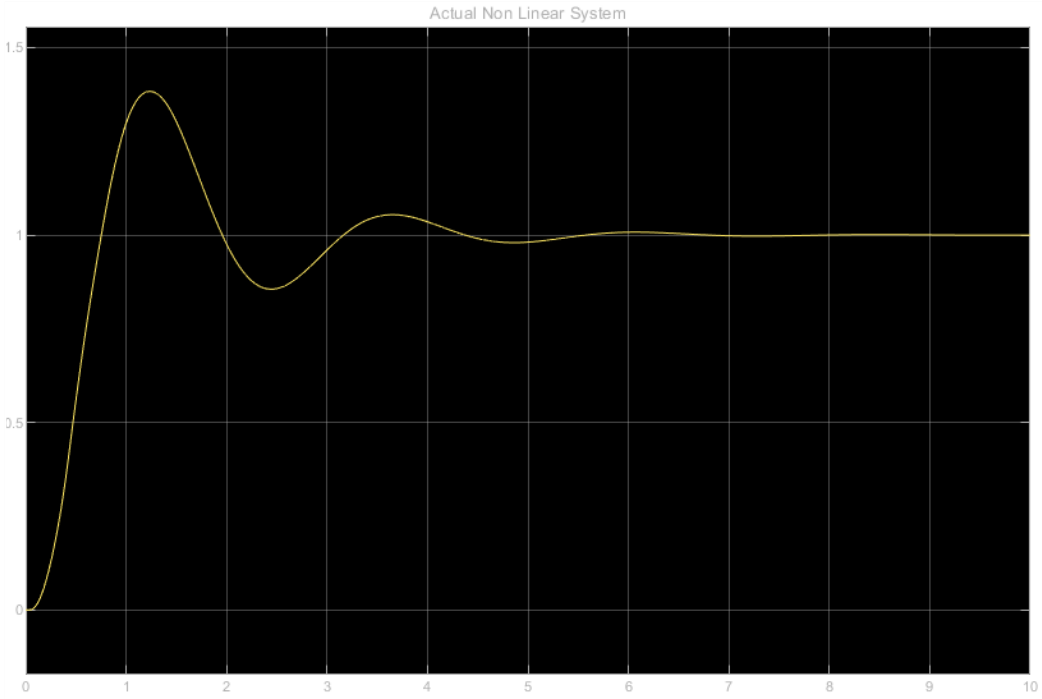


نتایج شبیه سازی برای ورودی برابر ۱۰:

برای سیستم شناسایی شده:



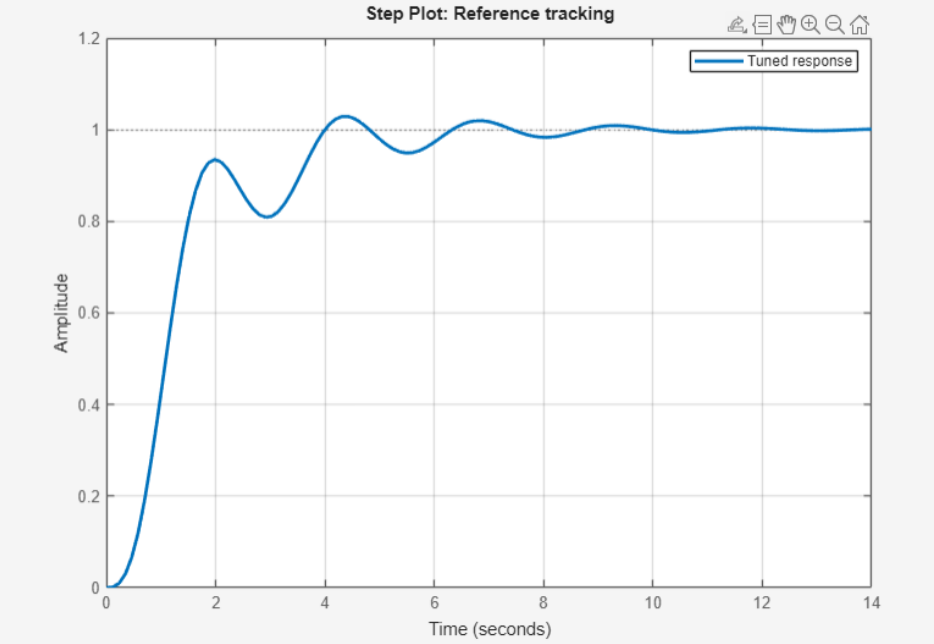
برای سیستم اصلی غیر خطی:



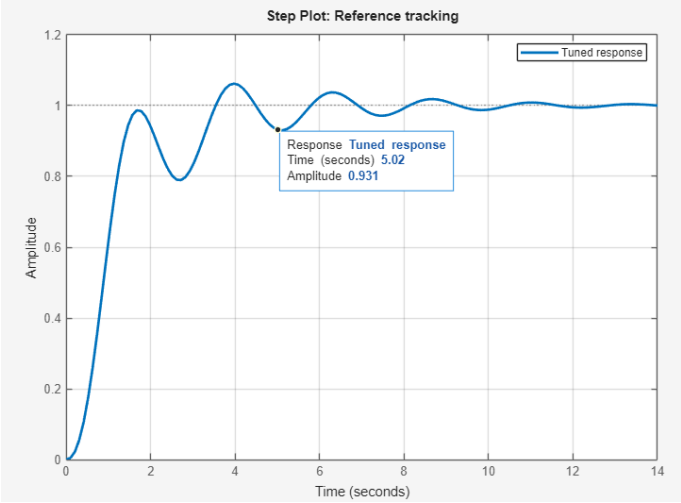
5. طراحی کنترل کننده PID با استفاده از pidTuner متلب برای سیستم با مشخصات فراجهش کمتر از ۲۰ درصد و زمان نشست کمتر از ۴ ثانیه بر روی سیستم شناسایی شده

با استفاده از pidTuner متلب به طراحی کنترل کننده میپردازیم:

ابتدا pidTuner خود کنترل کننده ای ساده طراحی کرده و این نمودار پاسخ پله را برای ما ایجاد میکند:



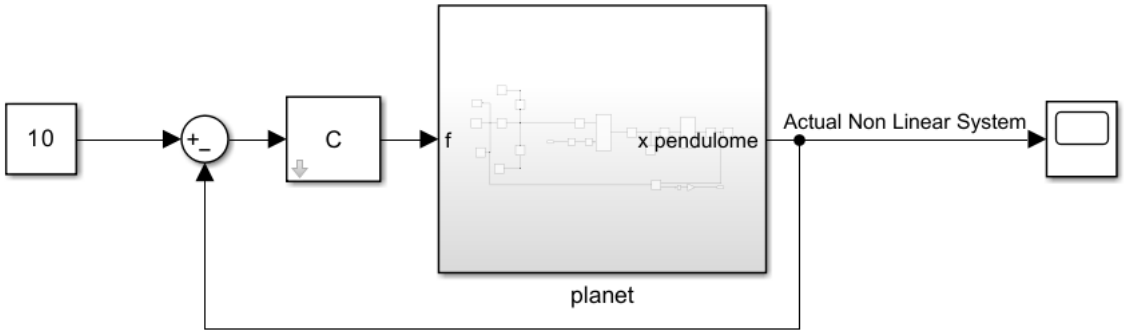
ولی از آنجایی که مشخصات خواسته شده را ندارد، با تنظیم بهتر میتوان به کنترل کننده زیر رسید و آن را ذخیره نمود:

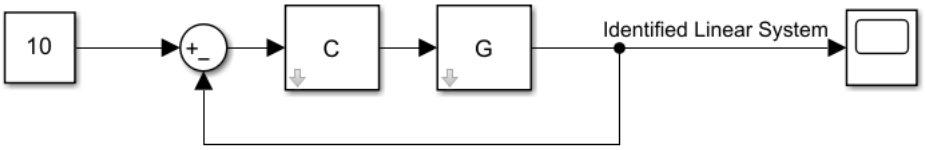


همانطور که میبینیم کنترل کننده طراحی شده فراجهش کمی دارد و همچنین سیستم در حدود ۳ ثانیه به ناحیه ی ۱۰ درصد مقدار نهایی وارد میشود. هرچند سیگنال خروجی مقداری نوسان قبل از نشست دارد ولی هم ورودی را دنبال میکند و هم مشخصات گفته شده را داراست.

6. تست کردن کنترل کننده طراحی شده در قسمت ۵ بر روی سیستم اصلی و سیستم شناسایی شده

برای این کار کنترل کننده طراحی شده در قسمت ۵ و سیستم شناسایی شده را از قبل لود میکنیم و سیستم سیمیولینک خود را به صورت زیر آماده میکنیم:

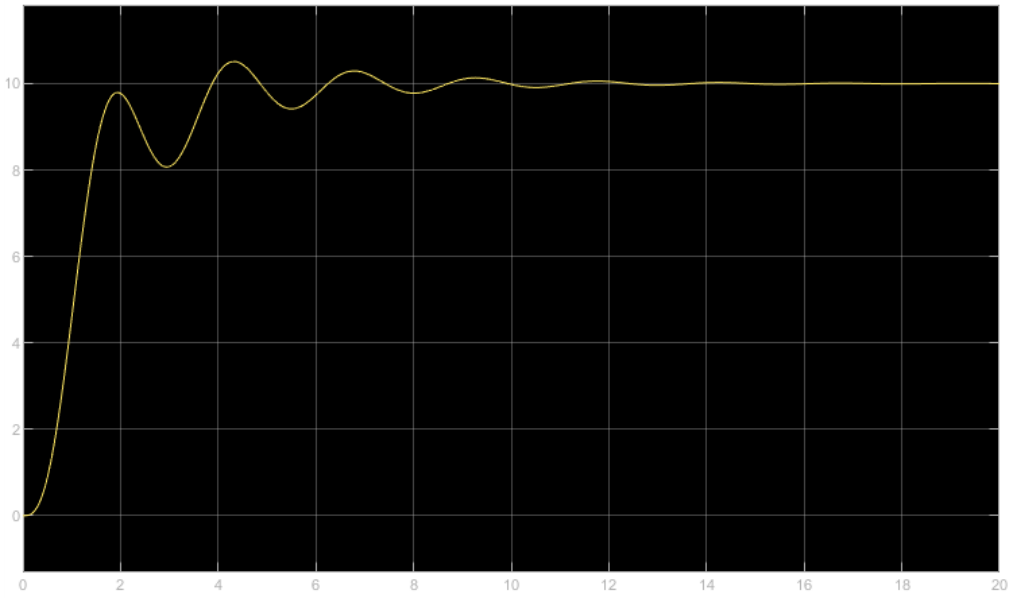




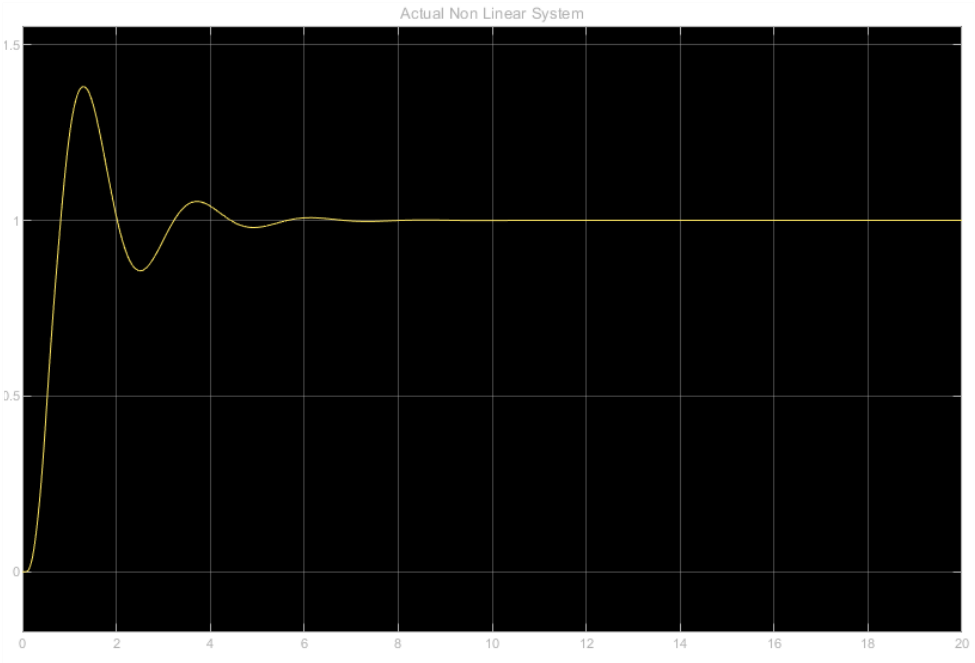
توجه داشته باشید که بلوک C همان کنترل کننده PID ما می باشد.

نتایج شبیه سازی برای ورودی برابر ۱۰:

برای سیستم شناسایی شده:

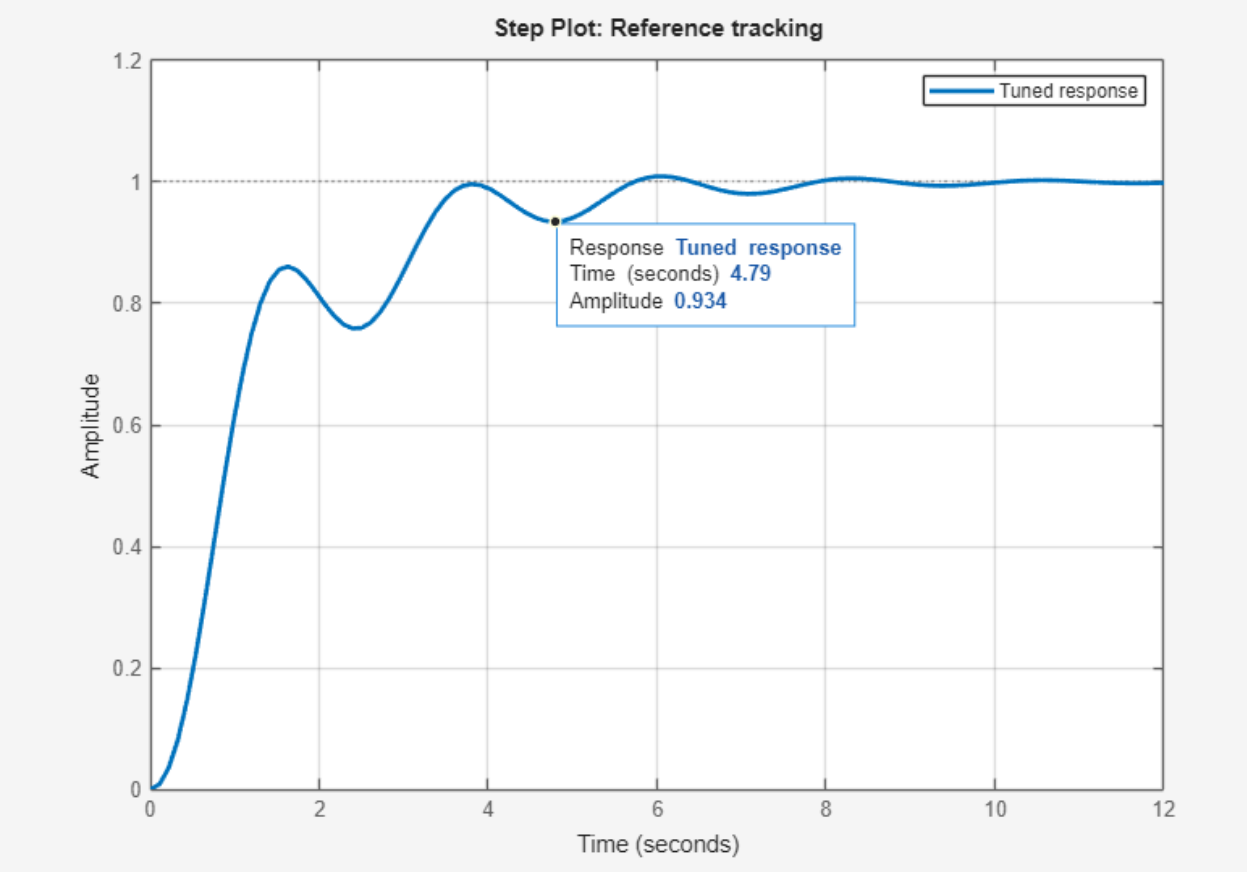


برای سیستم اصلی غیر خطی:



7 . تغییر ضرایب کنترل کننده برای زمان نشست کمتر از ۵ ثانیه

دوباره با استفاده از pidTuner متلب به طراحی کنترل کننده میپردازیم و نتیجه نهایی به شکل زیر میباشد:



دقت باید داشت که همه تفاوت های خروجی کنترل کننده ها بر روی سیستم شناسایی شده و سیستم واقعی بنا به یکی از دلایل زیر میباشد:

1 . استفاده نشدن کنترل کننده روی نقطه ی کاری که سیستم خطی سازی شده
2 . خطای مدل خطی